Engineer

HTML

Sprache zur Beschreibung der Struktur einer Webseite

Was es ist

HTML, kurz für HyperText Markup Language und noch immer weithin HTML5 genannt, ist das Klartextformat, in dem jede Webseite ankommt: Inhalt in Tags, die angeben, was jeder Teil ist. So kann der Browser eine Überschrift von einem Absatz, einen Link von einem Listeneintrag und ein Formularfeld von seiner Beschriftung unterscheiden.

Ein Browser liest diesen Text von oben nach unten und baut daraus einen Elementbaum, das Document Object Model. Diesen Baum gestalten Stylesheets, verändern Skripte und geben Screenreader wieder. Die Arbeitsteilung im Web ist alt und gilt weiterhin: HTML sagt, was etwas ist, CSS, wie es aussieht, und JavaScript, was es tut, wenn jemand damit interagiert. Ein Element wie `<nav>`, `<button>` oder `<h2>` trägt eine Bedeutung, die jeder dieser Leser nutzen kann. Ein echtes `<button>` erhält Tastaturfokus, eine von assistiver Technik angesagte Rolle und die Aktivierung durch Eingabe- oder Leertaste, ganz ohne zusätzliche Codezeile.

Die Sprache begann als Möglichkeit, wissenschaftliche Dokumente semantisch zu beschreiben, und wurde zur Grundlage fast jeder Anwendung mit Weboberfläche. Ihre ersten fünf Jahre, 1990–1995, verbrachte sie am CERN und danach bei der IETF; HTML 3.2 wurde 1997 beim W3C fertiggestellt, kurz darauf folgte HTML4. Jahrelang führten W3C und WHATWG auseinanderlaufende Texte, bis sie 2019 vereinbarten, an einem einzigen zu arbeiten: dem HTML Living Standard, den die WHATWG pflegt und jede große Browserengine implementiert. Er trägt keine Versionsnummer. Auf die Frage „Ist das HTML5?“ antwortet der Standard mit einem kurzen Ja und bezeichnet den Begriff als Schlagwort für die moderne Webplattform statt als Versionsbezeichnung.

Wo steht HTML neben den Formaten und Werkzeugen, mit denen es verglichen wird? Markdown ist eine Kurzschreibweise, die ein Buildwerkzeug in HTML umwandelt. PDF legt das Layout auf eine gedruckte Seite fest, während HTML sich an den jeweiligen Bildschirm anpasst. XHTML ist dasselbe Vokabular nach den strengeren Regeln von XML. JSX, Jinja und Hugo-Vorlagen sind Wege, HTML zu erzeugen; was davon beim Browser ankommt, bleibt HTML. Ein Team, das eine von Hand gebaute Website gegen einen Websitebaukasten abwägt, entscheidet, wer das Markup schreibt, niemals, ob es welches gibt.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist HTML?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Warum es wichtig ist

Wenige Teams entscheiden sich bewusst für HTML. Es kommt als das an, was ihr Framework, Seitenbaukasten oder Inhaltssystem gerade ausgibt. Tatsächlich entscheiden sie, ob diese Ausgabe ein von jemandem geprüftes Arbeitsergebnis oder ein von niemandem gelesenes Nebenprodukt ist. Die Folgen zeigen sich an drei Stellen.

Zuerst geht es darum, wer die Seite nutzen kann. Assistive Technik liest den Baum, nicht das Design. Screenreadernutzer bewegen sich anhand von Überschriften und Seitenregionen durch eine Seite; Tastaturnutzer springen mit der Tabulatortaste von einem Link oder Bedienelement zum nächsten. Ein gestaltetes `<div>` mit Klickbehandlung sieht für einen sehenden Mausnutzer wie eine Schaltfläche aus, ist für Tastatur- und Screenreadernutzer aber einfacher Text. Barrierefreiheitsanforderungen wie WCAG beziehen sich größtenteils auf die vom Markup offengelegten Informationen. Sie nach dem Start zu erfüllen bedeutet daher häufig, Vorlagen neu zu schreiben, statt Farben anzupassen.

An zweiter Stelle steht die Lebensdauer der Inhalte. Browser öffnen noch immer Seiten aus den 1990er-Jahren, weil Rückwärtskompatibilität zu den Grundsätzen der Standardpflege gehört. Frameworks wechseln alle paar Jahre. Inhalte in

sauberem, gültigem HTML lassen sich in ein anderes System übertragen, archivieren, in andere Formate umwandeln oder neu gestalten, ohne sie umzuschreiben. In verschachtelten Hüllen eines Seitenbaukastens vergrabene Inhalte müssen dagegen meist von Hand herauskopiert werden.

An dritter Stelle stehen Geschwindigkeit und Reichweite. Eine vom Server vollständig gesendete Seite kann bereits beim Eintreffen der Bytes dargestellt werden. Eine Anwendung, die eine fast leere Hülle sendet und das Dokument mit JavaScript aufbaut, lässt jeden Leser auf das Skript warten. Alles, was die Seite ohne dessen Ausführung liest, von Linkvorschauen bis zu vielen Crawlern, sieht nur die Hülle. Bei einem Produkt hinter einer Anmeldung, bei dem die Interaktion im Mittelpunkt steht, kann sich dieser Kompromiss lohnen. Für Marketingseiten, Dokumentation und alles, was über die Suche gefunden werden soll, ist vollständiges HTML die günstigere Standardlösung.

Der Preis aller drei Vorteile ist derselbe und gering: jemand im Team, der HTML genauso sorgfältig liest wie Code. Semantisches Markup kostet fast nichts, wenn es zuerst geschrieben wird, und sehr viel, wenn es nachgerüstet wird.

Wie es in der Praxis aussieht

Diese Website ist das naheliegendste Beispiel. Ihre Credits-Seite führt auf, was den Browser eines Besuchers erreicht: HTML, ein Stylesheet, lokale Webfonts und Bilder, ohne ausführbares JavaScript. Jede Seite wird vollständig als Datei auf einen Datenträger geschrieben, bevor jemand sie anfordert, statt pro Anfrage aus einer Vorlage zusammengesetzt zu werden. Die Qualitätsprüfung der Website validiert dieses HTML mit vnu und behandelt jede Diagnose als Fehler. Dieselbe Seite verlinkt den W3C Markup Validator, der diese Prüfengine gegen den Living Standard ausführt. Leser können die Aussage damit überprüfen, statt ihr vertrauen zu müssen.

Frühere Kundenarbeit näherte sich HTML von der anderen Seite. Bei dreißig aus fertigen visuellen Entwürfen erstellten Websites wurden die Photoshop-Dateien von Hand in HTML umgesetzt. Der Anspruch hatte zwei Seiten: Abstände, Typografie und Details mussten der Absicht des Designers entsprechen, und die Website musste flexibel genug bleiben, um auch nach der ersten Inhaltsänderung noch zu funktionieren. Einen Entwurf nachzubauen ist die eine Hälfte dieser Arbeit; Markup, das die nächste Änderung übersteht, ist die andere.

HTML als kanonische Form zu behalten lässt außerdem Raum, andere Formate daneben zu veröffentlichen. Bei der Spiegelung der Website auf Gemini und Gopher erzeugte derselbe Build beim dokumentierten Implementierungsstand aus derselben Quelle 777 Gemini-Dokumente und 777 Gopher-Menüs. Ein separater Buildvergleich maß beim Hinzufügen dieser Formate null veränderte HTML-Bytes. Dieses Ergebnis beschreibt keine späteren redaktionellen oder visuellen Änderungen.

Wo es schiefgeht

Das Layout verschiebt sich auf eine Weise, die kein Stylesheet erklärt. Eine Seite, die ohne `<!DOCTYPE html>` oder mit einer alten, fehlerhaften Variante beginnt, versetzt den Browser in den Quirks-Modus, der ältere Browser nachahmt. Prozentuale Höhen, Tabellenzellen und Zeilenhöhen folgen dann altem Verhalten, das von den CSS-Spezifikationen abweicht. Das Ergebnis unterscheidet sich von derselben Seite im Standardmodus. Die Lösung ist eine Zeile am Anfang jeder Vorlage.

Fehlerhaftes Markup wird dargestellt, aber der Baum sieht anders aus. Der HTML-Parser fängt jeden Fehler auf festgelegte Weise ab, in jedem Browser gleich; deshalb sind Fehler selten auf dem Bildschirm sichtbar. Sie treten erst später hervor, wenn ein Selektor oder Skript ein Element an einer anderen Stelle sucht, als der Parser es abgelegt hat. Der klassische Fall ist ein Blockelement in einem Absatz:

```
<p>Opening line
  <div>A block placed inside it</div>
</p>
```

Der Parser schließt den Absatz vor dem `<div>` und macht dann aus dem verwaisten `</p>` einen zweiten, leeren Absatz. Führen Sie im Build einen Validator aus und lassen Sie den Build bei dessen Fehlern scheitern. Die Toleranz des Browsers ist ein Wiederherstellungsmechanismus; sie prüft nichts auf Qualität.

Jedes Bedienelement wird zum `div`. Anklickbare `div`- und `span`-Elemente, nach ihrer Größe gewählte Überschriften, Tiefstellung zum Verkleinern von Text: Alles sieht richtig aus und vermittelt assistiver Technik etwas Falsches. Tastaturnutzer überspringen Bedienelemente, die nie Fokus erhalten, und

Screenreadernutzer verlieren die Überschriften und Seitenregionen, anhand derer sie navigieren. Ohne `<main>` oder `<h1>` müssen sie jedes Mal von oben lesen. Nutzen Sie das Element, das die Aufgabe benennt, etwa `button`, `a`, `h2`, `main` oder `nav`, und ändern Sie sein Aussehen mit CSS.

Die Seite bleibt leer, bis ihr Skript läuft. Eine im Client gerenderte Website sendet kaum mehr als `<div id="app"></div>` und baut das Dokument im Browser auf. Auf einem langsamen Smartphone wartet der Leser; bei einem Skriptfehler oder einer Blockierung durch einen Filter bleibt die Seite dauerhaft leer. Rendern Sie das Dokument auf dem Server oder beim Build und ergänzen Sie interaktive Funktionen auf der Grundlage von HTML, das bereits für sich sinnvoll ist.

Weiterführende Lektüre

- HTML Living Standard — Die von Browsern implementierte Spezifikation, von der WHATWG gepflegt und fortlaufend aktualisiert.
- Einführung in den HTML-Standard — Die Geschichte vom CERN bis zur Vereinbarung von 2019 und die Antwort des Standards selbst auf die HTML5-Frage.
- HTML-Dokumente parsen — Die allen Browsern gemeinsame Fehlerbehandlung, Regel für Regel beschrieben.
- Die HTML-Syntax — Doctype, Elemente und Attribute sowie die Anforderungen an ein konformes Dokument.
- Der Nu Html Checker — Die Prüfengine hinter vnu und dem W3C-Markupvalidator mit ihrer Nutzung auf der Kommandozeile.
- DOM-Standard — Der vom Parser erzeugte Baum und die Schnittstellen, mit denen Skripte ihn verändern.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist HTML?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/html/