Engineer

Responsives Webdesign

Layout, das sich an den jeweiligen Bildschirm anpasst

Was es ist

Responsives Webdesign bedeutet, eine Seite so zu bauen, dass sich ihr Layout an den jeweiligen Bildschirm anpasst, vom schmalen, hochkant gehaltenen Mobiltelefon bis zum breiten Monitor, statt für jede Geräteart eine eigene Website zu bauen. Die längere englische Bezeichnung responsive web design ist der Titel des Artikels, der dem Ansatz seinen Namen gab: Ethan Marcottes Beitrag in A List Apart vom 25. Mai 2010. Er übernahm die Idee aus einer von ihm als neu entstehend beschriebenen Disziplin namens responsive Architektur, bei der sich Wände und Installationen bewegen und ihre Form ändern, wenn Menschen sich nähern.

Marcotte nannte fließende Raster, flexible Bilder und Media Queries „die drei technischen Zutaten für responsives Webdesign“. Ein fließendes Raster bemisst seine Spalten als Anteile des verfügbaren Platzes statt in festen Pixeln. Dadurch wächst und schrumpft das Layout mit dem Fenster. Ein flexibles Bild darf niemals breiter als die Spalte werden, die es enthält. Eine Media Query lässt ein Stylesheet dort umschalten, wo Proportionen allein nicht mehr funktionieren: Werden etwa drei Spalten zum Lesen zu schmal, wechselt die Seite zu einer.

Die Spezifikation Media Queries Level 4, ein W3C Candidate Recommendation Draft vom 19. Februar 2026, definiert eine Media Query als logischen Ausdruck mit zwei möglichen Werten, wahr oder falsch: einen Test einer Eigenschaft des Browsers oder Geräts, das das Dokument anzeigt. Der Test, auf den sich responsive Layouts stützen, ist `width`. Bei einem Bildschirm ist das die Breite des Viewports einschließlich der Bildlaufleiste. Längen in einer Abfrage sind dieselben CSS-Längen wie überall sonst. Ein Pixel ist dort also ein CSS-Pixel und kein physischer, und ein `em` wird an der anfänglichen Schriftgröße des Browsers gemessen. Diese kann der Leser ändern; maßgeblich ist niemals eine von der Seite gesetzte Größe.

Auf einem Mobiltelefon funktioniert das nur, wenn die Seite darum bittet. Ohne Viewport-Metaelement legen mobile Browser eine Seite wie für ein Desktopfenster aus, typischerweise 980 bis 1.024 CSS-Pixel breit, und verkleinern anschließend das Ergebnis. Jede Media Query sieht deshalb einen Desktop. Der Entwurf CSS Viewport bezeichnet das als Ausgleich für Seiten, die nur für große Bildschirme entworfen wurden, und als unnötig für Seiten, die auch auf kleinen gut funktionieren. Mit der Zeile `width=device-width, initial-scale=1` im folgenden Beispiel sagt eine Seite, dass sie zur zweiten Gruppe gehört.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist responsives Webdesign?“ 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

Marcottes Argument betraf die Wartung ebenso wie das Aussehen. Statt voneinander getrennte Entwürfe an immer mehr Geräte anzupassen, wollte er sie als Facetten derselben Nutzungserfahrung behandeln. Eine Seite, eine Adresse und ein Stylesheet sorgen dafür, dass eine Korrektur alle Leser gleichzeitig erreicht. Auch eine neue Bildschirmgröße wird von derselben Website bedient, weil das Layout von Anfang an nicht für eine bestimmte Größe geschrieben wurde.

Den Zoom unterschätzen viele Teams als Grund für responsives Design. Beim Hineinzoomen wird das CSS-Pixel größer. Der in CSS-Pixeln gemessene Viewport wird dadurch schmaler. Die Web Content Accessibility Guidelines 2.2, eine W3C Recommendation vom 12. Dezember 2024, verlangen auf Stufe AA im Erfolgskriterium 1.4.10 Umfluss (Reflow), dass Inhalte bei einer verfügbaren Breite von 320 CSS-Pixeln ohne Scrollen in zwei Richtungen nutzbar sind. Sie erläutern ausdrücklich, dass ein 1.280 CSS-Pixel breites Fenster bei 400 % Zoom genau diese Breite erhält. Bei dieser Ausgangsbreite sieht jede Media Query einen Leser, der seinen Laptopbrowser auf 400 % zoomt, als Nutzer eines 320 Pixel breiten Viewports. Die W3C-Erklärung des Kriteriums stellt fest, dass

Websites häufig responsives Design für kleinere Viewports einsetzen. Inhalte, die in einen kleinen Viewport passen, helfen deshalb auch Menschen, die auf größeren Geräten hineinzoomen, nicht nur Menschen mit einem Mobiltelefon.

Der dritte Grund sind Bytes. Der HTML Standard nennt drei verschiedene Gründe, mehrere Dateien für ein Bild anzubieten: die Auswahl nach Viewport, die gestalterische Bildauswahl und die Auswahl nach Bildformat. Ein Foto über die Breite eines Mobiltelefons benötigt viel weniger Pixel als dasselbe Foto auf einem großen Monitor. Mit `srcset` und `sizes` kann der Browser ohne Skript die kleinere Datei abrufen.

Eine Seite kann auf mehr als die Breite reagieren. Media Queries Level 4 beschreibt auch das primäre Zeigegerät: `(pointer: coarse)` passt zu einem Gerät mit begrenzter Genauigkeit, etwa einem Touchscreen. `(hover: none)` passt zu einem primären Zeigegerät, dessen Zeiger nicht ohne Aktivierung über einem Element positioniert werden kann, oder wenn überhaupt kein Zeigegerät vorhanden ist. Schon Marcottes Artikel schlug vor, Links auf kleinen Bildschirmen mit Media Queries größere Zielflächen zu geben.

Wie es in der Praxis aussieht

Eine Kartenliste mit einer Spalte auf dem Mobiltelefon, zwei auf einem quer gehaltenen Tablet und drei auf einem breiten Bildschirm beginnt mit der Viewport-Zeile und einem Bild, das der Browser selbst auswählen kann:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />

<img
  src="pool-400.jpg"
  srcset="pool-400.jpg 400w, pool-800.jpg 800w, pool-1600.jpg 1600w"
  sizes="(max-width: 40em) 100vw, (max-width: 64em) 50vw, 33vw"
  width="1600"
  height="900"
  alt="A tide pool at low water"
/>
```

Das Stylesheet enthält das Layout:

```
.cards {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1rem;
}

@media (width >= 40em) {
  .cards {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (width >= 64em) {
  .cards {
    grid-template-columns: repeat(3, 1fr);
  }
}

.cards img {
  max-width: 100%;
  height: auto;
}
```

Das Raster ist der fließende Teil: `1fr`-Spalten teilen die vorhandene Breite untereinander auf, nichts hat eine feste Größe. Die beiden Media Queries schalten das Layout um. Da sie in `em` geschrieben sind, folgen sie der Standardschriftgröße des Lesers. Wer eine größere Standardschrift eingestellt hat, bekommt bei Fensterbreiten noch eine Spalte, bei denen andere bereits zwei sehen. Genau diesen Platz benötigen die längeren Zeilen. Die letzte Regel hält jedes Bild innerhalb seiner Spalte.

Das Attribut `sizes` teilt dem Browser schon vor der Layoutberechnung mit, wie breit das Bild angezeigt wird: Die erste zutreffende Bedingung gewinnt, und der letzte Wert gilt, wenn keine zutrifft. Bei einem 39em breiten Viewport trifft die erste Bedingung zu, das Bild zählt also als 39em breit. Bei 41em trifft die zweite zu, und das Bild zählt als halb so breit wie der Viewport, also 20,5em. Das etwas breitere Fenster benötigt die kleinere Datei, weil sich das Layout geändert hat. Der HTML Standard führt denselben Effekt in seinem eigenen Beispiel vor. Aus diesen Zahlen und den `w`-Deskriptoren in `srcset` berechnet der Browser, welche Dichte jede Datei auf diesem Bildschirm hätte. Er darf jede davon wählen und dabei Pixeldichte, Zoomstufe und möglicherweise das Netzwerk berücksichtigen. Mit den Attributen `width` und `height` kann er den Platz für das Bild schon vor dessen Download reservieren.

Ein Portfolio-Eintrag zur Beschleunigung der Browserprüfungen dieser Website zeigt denselben Gedanken als Test. Bei dieser Arbeit im August 2026 erhielten die Prüfungen, die einen Browser ohne Oberfläche steuern, eine gemeinsame Tabelle mit zweiundzwanzig Fensterformaten. Die Formate wurden aus allen tatsächlich im Stylesheet vorkommenden Media Queries abgeleitet. Die Laufzeit der Layoutprüfung, die die Seite in jedem vom Design vorgesehenen Fensterformat betrachtet, stieg dadurch von 95 auf 405 Sekunden. Die Formate stammen aus dem Stylesheet, nicht aus einer Geräteliste. Deshalb werden die Seiten an den tatsächlich vorhandenen Breakpoints geprüft.

Wo es schiefgeht

Breakpoints fragen nach dem Gerät. Marcottes Beispiele von 2010 fragten `max-device-width` ab, die Breite des gesamten Bildschirms. Media Queries Level 4 erklärt `device-width`, `device-height` und `device-aspect-ratio` inzwischen für veraltet und schreibt vor, sie nicht mehr zu verwenden. Die Spezifikation weist darauf hin, dass sie oft zur Erkennung mobiler Geräte dienen, obwohl sie nichts darüber sagen, wie viel Platz die Seite tatsächlich hat. Fragen Sie stattdessen `width` ab und setzen Sie jeden Breakpoint dorthin, wo der Inhalt nicht mehr funktioniert, nicht an die Bildschirmkante eines verbreiteten Mobiltelefons.

Zwischen zwei Breakpoints bleibt eine Lücke. Ein Stylesheet mit `(max-width: 320px)` für ein Layout und `(min-width: 321px)` für das nächste sieht vollständig aus. Die Spezifikation weist aber darauf hin, dass ein Viewport mit einer Breite strikt zwischen 320 und 321 Pixeln zu keinem von beiden passt. Nicht ganzzahlige Breiten kommen tatsächlich vor, etwa durch nicht ganzzahlige Pixeldichten oder Zoom. Bei solchen Breiten gilt keiner der beiden Stilblöcke. Verwenden Sie die Bereichssyntax `(width <= 320px)` und `(width > 320px)`, damit jede Breite in genau einen Bereich fällt.

Der Zoom ist abgeschaltet. Eine Viewport-Zeile mit `user-scalable=no` oder einem `maximum-scale` unter 2 besteht die W3C-Testregel zum Erfolgskriterium 1.4.4 Textgröße ändern (Resize Text) nicht. Dieses verlangt, Text auf 200 Prozent vergrößern zu können. Die Regel merkt an, dass Desktopbrowser das Element ignorieren. Die meisten aktuellen mobilen Browser beachten es entweder von Anfang an nicht oder bieten eine Barrierefreiheitseinstellung, die Zoom trotzdem erlaubt. Betroffen sind deshalb noch Menschen mit älteren mobilen Browsern. Lassen Sie beide Eigenschaften weg und sorgen Sie stattdessen dafür, dass das Layout mit Zoom nutzbar bleibt. Eine kleinere Schrift an einem

schmalen Breakpoint erfordert dieselbe Sorgfalt: Die W3C-Erklärung zum Umfluss erlaubt sie nur, wenn Leser den Text auf irgendeinem Weg immer noch auf die doppelte normale Größe bringen können.

Bilder ignorieren das Layout. Bei `w`-Deskriptoren in `srcset` ohne `sizes` nimmt der Browser laut HTML Standard an, das Bild fülle den gesamten Viewport. Für ein Bild auf einem Drittel des Bildschirms lädt er deshalb möglicherweise eine etwa dreimal so breite Datei wie nötig. Prozentangaben sind in `sizes` nicht erlaubt; verwenden Sie `vw`. Wenn `srcset` mit `w`-Deskriptoren arbeitet, ignorieren Browser, die `srcset` verstehen, das Attribut `src`. Nur `src` zu ändern, ändert für die meisten Leser also nichts. Schreiben Sie `sizes` passend zum CSS und ändern Sie beides gemeinsam, wenn sich das Layout ändert. Die Bildauswahl per Skript löst das nicht: Der Standard warnt, dass dadurch das Bild zweimal geladen werden kann oder bei abgeschalteten Skripten gar nicht.

Ein breites Element schiebt die ganze Seite zur Seite. Eine Datentabelle, eine lange Codezeile oder eine Einbettung mit fester Breite kann breiter als 320 CSS-Pixel sein und die gesamte Seite horizontal scrollen lassen. Dann muss jeder Absatz durch Hin- und Herscrollen gelesen werden. Die W3C-Erklärung zum Umfluss nimmt Inhalte aus, die zwei Dimensionen benötigen, etwa Datentabellen, Karten und Videos, nicht aber die Absätze darum herum. Geben Sie der Tabelle einen eigenen, mit `overflow-x: auto` separat scrollbaren Container. Bilder und Videos bekommen `max-width: 100%`, damit nur der Teil scrollt, der es benötigt.

Ein schmaler Bildschirm wird mit einem Touchscreen verwechselt. Ein auf 400 % gezoomter Laptop ist schmal und hat eine Maus; ein großes Tablet ist breit und unterstützt überhaupt kein Hover. Das Layout gehört zu `width`, Zielflächen und Hover-Effekte gehören zu `pointer` und `hover`. Media Queries Level 4 empfiehlt, auch `any-pointer` und `any-hover` zu berücksichtigen, weil ein Gerät mehrere Arten der Zeigerbedienung haben kann.

Weiterführende Lektüre

- Responsive Web Design — Ethan Marcottes Artikel vom 25. Mai 2010, der den Ansatz und seine drei Zutaten benannte.
- Media Queries Level 4 — Das Merkmal `width`, Bereichssyntax, relative Einheiten, Zeigegeräte und Hover sowie warum `device-width` als veraltet gilt.

- HTML Standard: Bilder — `srcset`, `sizes`, die drei Gründe für mehrere Dateien und das ausgearbeitete Beispiel mit 29em und 32em.
- Erfolgskriterium 1.4.10 verstehen: Reflow — Warum 320 CSS-Pixel, wie Zoom damit zusammenhängt und welche Inhalte ausgenommen sind.
- Meta-Viewport erlaubt Zoom — Die W3C-Testregel für `user-scalable` und `maximum-scale` mit ihrem Hinweis zur Browserunterstützung.
- CSS Viewport Module Level 1 — Das Viewport-Metaelement, seine Eigenschaften und warum mobile Browser eine Desktopbreite annehmen.

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 responsives Webdesign?“ 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/responsive-design/