Engineer

Viewport

der Bereich, in dem ein Browser eine Seite darstellt

Was es ist

Ein Viewport ist das Rechteck, in dem ein Browser ein Dokument anordnet und zeichnet: am Bildschirm der Bereich innerhalb der browserseitigen Werkzeugleisten, auf Papier der Seitenbereich jedes Blatts. CSS bezieht sich ständig darauf. Der initiale umschließende Block, also die Box, in der das Wurzelelement angeordnet wird, übernimmt seine Größe. Eine Media Query für `width` oder `height` fragt nach seinen Maßen, und die Einheiten `vw` und `vh` sind Hundertstel davon. Auf einem Smartphone braucht das Wort Viewport, ebenso wie sein Plural Viewports, eine nähere Bestimmung. Die Spezifikationen kennen mehrere: Layout-Viewport, visueller Viewport sowie kleinen, großen und dynamischen Viewport.

Am Desktop sind Viewport und Fenster fast dasselbe. Auf dem Smartphone fallen sie auseinander. Weil so viele Dokumente für Desktopfenster entworfen wurden, ordnen mobile Browser eine Seite ohne weitere Vorgaben mit einer festen, desktopähnlichen Breite an. Laut CSS-Viewport-Spezifikation sind das gewöhnlich 980 bis 1.024 CSS-Pixel. Anschließend verkleinern sie das Ergebnis auf die Displaybreite. Ein Element im Kopf der Seite verhindert das. Es versteht `width`, `height`, drei Skalierungsgrenzen, `user-scalable` und `interactive-widget`:

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

Beim Zoomen teilt sich der Viewport in zwei. Der Layout-Viewport ist die Bezugsfläche für das CSS-Layout. Beim Vergrößern mit zwei Fingern bleibt er unverändert, also bleiben auch die Zeilenumbrüche bestehen. Der visuelle Viewport ist der Teil der Seite, der gerade auf dem Display zu sehen ist: Er wird beim Vergrößern kleiner und wandert beim Verschieben mit. Eine Bildschirmtastatur verkleinert standardmäßig nur den visuellen Viewport.

`interactive-widget=resizes-content` fordert den Browser auf, auch das Layout anzupassen; `overlays-content` verlangt, keinen der beiden in der Größe zu verändern.

Hinzu kommen die ein- und ausblendbaren Bedienelemente des mobilen Browsers. Beim Laden nimmt die Adressleiste Platz ein; beim Scrollen zieht sie sich zurück. Dadurch ändert sich die sichtbare Höhe, ohne dass die Seite selbst verändert wurde. CSS Values 4 unterscheidet dafür drei Größen. Der große Viewport setzt voraus, dass alle einziehbaren Werkzeugleisten eingezogen sind; der kleine, dass alle angezeigt werden; der dynamische folgt dem jeweiligen Zustand. Die Einheiten tragen entsprechende Präfixe: `lvh`, `svh` und `dvh`, mit passenden Breitenvarianten. Die einfachen Einheiten `vh` und `vw` beziehen sich auf den großen Viewport. Eine Bedienoberfläche, die den Inhalt lediglich überlagert, statt ihn zu verschieben, etwa die Bildschirmtastatur, ändert keine dieser Längen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Viewport?“ 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

Responsives Design baut auf dem Viewport auf. Ein Breiten-Breakpoint im Stylesheet ist eine Frage nach seiner Breite. Fehlt das `meta`-Element, wird eine Seite deshalb auch auf dem Smartphone in Desktopbreite angeordnet und so weit verkleinert, dass der Text unlesbar wird, ganz gleich, wie sorgfältig ihre Media Queries geschrieben sind.

Die Höhe ist die schwierige Achse. Die Breite kann sich beim Drehen oder beim Ändern der Fenstergröße ändern; die Höhe kann sich zusätzlich verändern, wenn sich die Werkzeugleiste ein- oder ausblendet, die Tastatur aufgeht oder das Smartphone gedreht wird. Alles, was sich nach der Bildschirmhöhe richtet, muss daher festlegen, welche Höhe gemeint ist. Mit dem kleinen Viewport passt der Inhalt auch bei sichtbaren einziehbaren Werkzeugleisten; nach deren Einziehen

bleibt ein leerer Streifen zurück. Der große füllt den Bildschirm, schiebt aber bei sichtbaren Werkzeugleisten Inhalt dahinter. Der dynamische berücksichtigt Änderungen dieses Zustands. Die Spezifikation selbst warnt, dass er die Größe von Inhalten während des Scrollens ändern kann, wodurch sich alles darunter verschiebt.

Bildschirme haben außerdem Ecken. Ein Smartphone mit abgerundeten Ecken oder einer Kameraaussparung ist kein Rechteck. Mit der Standardeinstellung `auto` bleibt die ganze Seite sichtbar. `viewport-fit=cover` fordert ein Rechteck um den gesamten physischen Bildschirm an und überträgt der Seite die Aufgabe, Text von der Aussparung fernzuhalten. Dafür dienen die Variablen `env(safe-area-inset-*)` . Auf einem rechteckigen Display müssen diese Abstände null sein. Eine einzige Padding-Regel funktioniert deshalb für beide Formen.

Auch feste Positionierung hängt am Viewport. Eine fest positionierte Box bezieht sich auf den Layout-Viewport, dessen Größe CSS Positioned Layout an den dynamischen Viewport anpasst. So bleiben Cookieleisten, modale Hintergründe und bildschirmfüllende Menüs stehen, während die Seite scrollt. Voraussetzung ist jeweils, dass kein Vorfahr diese Rolle übernommen hat. `transform` , `will-change` und `contain` können das bewirken.

Der Druck ist der Viewport, dem ein reines Bildschirm-Stylesheet nie begegnet. Auf Papier messen Viewport-Einheiten den Seitenbereich; eine Media Query für `width` fragt nach der Seitenbox, und ein fest positioniertes Element wird auf jeder Seite wiederholt.

Wie es in der Praxis aussieht

Die Seiten dieser Website deklarieren `width=device-width, initial-scale=1, viewport-fit=cover` . Sie werden damit in Smartphonebreite angeordnet und reichen bis an den Rand unter einer Displayaussparung. Die zweite Hälfte erledigt das Stylesheet. Seitliches Padding verwendet `max(1rem, env(safe-area-inset-left))` und dessen rechtes Gegenstück. Der Text hält dadurch ein rem Abstand zum Rand und rückt nur an Aussparungen weiter ein. Auf niedrigen Querformatbildschirmen steigt der Mindestabstand auf `max(2rem, 7vw, env(safe-area-inset-left))` : Ohne Aussparung sind die Insets null, und ein rem lag zu nah am Rand.

Bildschirmfüllende Abschnitte beziehen ihre Höhe aus einer gemeinsamen Variablen mit dem Wert `100dvh` und `100vh` als Rückfallwert. Der Kommentar daneben hält den Grund fest: `100vh` kann bei sichtbarer Adressleiste höher sein als der sichtbare Bereich und hatte den nächsten Abschnitt unter dem Aufmacher hervorschauen lassen. Zwei Elemente weichen bewusst davon ab. Das Markenzeichen im Seitenkopf verwendet statisches `vh`: Mit dem dynamischen Viewport würde es beim Einziehen der Browserleisten seine Größe ändern und mitten im Scrollen den Aufmacher neu umbrechen. Der feste Hintergrund auf Touchscreens ist `100lvh` hoch, damit er beim Ein- und Ausblenden der Adressleiste unverändert bleibt. Er liegt auf einer eigenen Compositing-Ebene, weil Touchbrowser ihn beim Nachlaufscrollen in helle und dunkle Bänder zerrissen hatten.

In einem früheren Startseitenlayout hatte jeder Dienstleistungsblock eine Mindesthöhe von `100dvh`. Solange sein Inhalt kürzer als der Bildschirm war, belegte er damit einen Bildschirm. Die im November 2026 geprüften Stylesheet-Kommentare halten einen Test bei 852 mal 393 Pixeln fest: Der Inhalt eines Blocks kam auf 412 Pixel, und die Blöcke stießen unmittelbar aneinander. In diesem Layout vergrößerte Padding innerhalb der `100dvh`, umgesetzt mit `box-sizing: border-box`, den Block auf einem hohen Bildschirm nicht und schuf erst dann Abstand, wenn der Inhalt überlief.

In der dokumentierten Ausbaustufe der Qualitätsprüfung vom August 2026 folgten die Tests den Media Queries. Die browsergestützte Qualitätsprüfung der Website weitete ihre Prüfungen auf eine gemeinsame Tabelle mit zweiundzwanzig Fensterformaten aus, abgeleitet aus sämtlichen Media Queries des Stylesheets. Allein die Layoutprüfung dauerte dadurch statt 95 nun 405 Sekunden.

Beim Drucken zeigten sich die Unterschiede zwischen den Rendering-Engines. Das Druck-Stylesheet hält fest, dass Viewport-Einheiten in einer Engine auf die Seitenbox und in einer anderen auf das Fenster bezogen wurden. Beim Drucken aus einem breiten Fenster fehlte in der zweiten daher ein Drittel jeder Zeile. Das Stylesheet des Seitenkopfs dokumentiert dieselbe Lücke bei Media Queries: Im Druck galten weiterhin Ausrichtung, Höhe und Breite des Fensters. Aus breiten und schmalen Fenstern gedruckte Blätter hatten deshalb um 47 Pixel unterschiedliche Seitenköpfe, bis `screen and` am Anfang der Abfrage die Regel auf Bildschirmfenster beschränkte.

Auch die Falle mit dem umschließenden Block ist dokumentiert. Die Arbeit an der Einstellung für reduzierte Bewegung zeigte, dass fünf verschiedene Eigenschaften ein Element jeweils zum umschließenden Block für viewportbezogen positionierte Elemente machen. Dadurch war eine Fläche zum Schließen einer Einblendung unbemerkt auf einen Bruchteil des Bildschirms geschrumpft.

Wo es schiefgeht

Das Smartphone zeigt eine winzige Desktopseite. Ohne das Viewport-Element `meta` ordnet der Browser die Seite in Desktopbreite an und verkleinert sie. Auch die Media Queries sehen diese Breite. Abhilfe schafft `width=device-width, initial-scale=1` ; Zoomen sollte erlaubt bleiben. `user-scalable=no` nimmt Lesern ihre eigene Möglichkeit, kleinen Text zu vergrößern.

Ein bildschirmfüllender Abschnitt reicht über den unteren Bildschirmrand. `100vh` meint den großen Viewport. Bei sichtbarer Adressleiste liegen der untere Rand des Abschnitts und dort platzierte Schaltflächen deshalb dahinter. `svh` hält ein Element von ausgeklappten Browserleisten frei; `dvh` eignet sich nur dort, wo Größenänderungen während des Scrollens vertretbar sind.

Ein bildschirmfüllendes Overlay schrumpft zu einer Box innerhalb der Seite. Ein Vorfahr mit `transform` , einem darauf verweisenden `will-change` oder bestimmten Containment-Einstellungen kann zum umschließenden Block für `position: fixed` werden. Das Overlay richtet sich dann nach diesem Vorfahren. Overlays sollten außerhalb transformierter Elemente gerendert werden.

Auf jeder Druckzeile fehlt das Ende. Viewport-Einheiten und auf Fenstermaße bezogene Media Queries wirken in den Druck hinein. Eine Engine, die sie gegen das Fenster auflöst, legt das Blatt für einen Bildschirm aus. Viewport-Einheiten gehören deshalb nicht in druckbare Elemente; für Bildschirmfenster bestimmte Abfragen brauchen davor `screen and` .

Weiterführende Lektüre

- CSS Viewport Module Level 1 — das Viewport-Element `meta` und die drei Verhaltensweisen von `interactive-widget` .
- CSS Values and Units Level 4: viewport-percentage lengths — kleiner, großer und dynamischer Viewport sowie sämtliche zugehörigen Einheiten.

- CSS Positioned Layout Level 3 — feste Positionierung, Eigenschaften, die ihren Bezugsblock ändern, und ihr Verhalten auf Papier.
- CSS Environment Variables Level 1 — die Safe-Area-Abstände und die Regel, die sie auf rechteckigen Bildschirmen null setzt.
- Media Queries Level 4 — `width`, `height` und `orientation`, am Bildschirm auf den Viewport und im Druck auf die Seitenbox bezogen.

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 ein Viewport?“ 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/viewport/