Engineer

Viewport

den del af en side et vindue faktisk viser

Hvad det er

En viewport er det rektangel, en browser lægger et dokument ud i og tegner det ind i: på en skærm området inden for vinduets egne værktøjslinjer, på papir sideområdet på hvert ark. CSS måler hele tiden mod det. Den indledende indeholdende blok, altså den boks, rodelementet lægges ud i, får dets størrelse, en medieforespørgsel på `width` eller `height` spørger til det, og enhederne `vw` og `vh` er hundredededele af det. På en telefon har ordet brug for et forled, og specifikationerne har flere at vælge imellem: layout, visuel, lille, stor og dynamisk.

På en computer er viewporten og vinduet næsten det samme. På en telefon skilles de. Fordi så mange dokumenter er lavet til skrivebordsvinduer, lægger mobilbrowsere en side uden instruktioner ud i en fast bredde, der efterligner et skrivebordsvindue, ifølge CSS Viewport-specifikationen typisk 980 til 1.024 CSS-pixel, og skalerer derefter resultatet ned, så det passer til glasset. En side melder sig ud med ét element i sit hoved, og elementet tager imod `width`, `height`, tre skalagrænser, `user-scalable` og `interactive-widget`:

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

Zoom deler viewporten i to. Layout-viewporten er den, CSS lægger siden ud efter, og knibezoom rører den ikke, så linjeskiftene bliver, hvor de er. Den visuelle viewport er den del af siden, der er på glasset netop nu: den bliver mindre, når læseren zoomer ind, og glider med, når læseren panorerer. Som standard skrumper et skærmtastatur kun den visuelle viewport. Med `interactive-widget=resizes-content` beder siden browseren om at ændre layoutet med, og med `overlays-content` om at lade begge være.

Telefoner har desuden browserbjælker, der glider væk. Adresselinjen tager plads, når siden indlæses, og trækker sig tilbage, når læseren ruller, så den synlige højde ændrer sig under en side, der står stille. CSS Values 4 giver tre størrelser til det. Den store viewport antager, at alle bjælker, der kan trækkes tilbage, er trukket tilbage, den lille antager, at de alle vises, og den dynamiske følger, hvad der faktisk er tilfældet. Enhederne bærer de samme forstavelser, `lvh`, `svh` og `dvh`, med tilsvarende bredder, og de almindelige `vh` og `vw` er defineret ud fra den store viewport. En brugerflade, der lægger sig over indholdet uden at flytte det, typisk skærmtastaturet, ændrer ingen af de længder.

Brug begrebet i en beslutning

Brug »Hvad er en viewport?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

Har styret full-stack-udviklingen af GIS-kort Kontakt os →

Hvorfor det betyder noget

Responsivt design hviler på viewporten. Et brudpunkt for bredden i et stylesheet er et spørgsmål om dens bredde, så en side uden `meta`-elementet bliver lagt ud i skrivebordsbredde på en telefon og skrumpet, til teksten er ulæselig, uanset hvor omhyggeligt medieforespørgslerne er skrevet.

Højden er den svære akse. Bredden kan ændre sig, når telefonen drejes, eller vinduet får en anden størrelse. Højden kan også ændre sig, når bjælken glider, når tastaturet åbner, og når telefonen vendes, så alt, der er dimensioneret efter skærmens højde, må sige, hvilken højde det mener. Med den lille viewport kan indholdet få plads, mens de bjælker, der kan trækkes tilbage, er fremme; den efterlader en stribe ubrugt plads, når bjælkerne trækkes tilbage. Den store fylder skærmen og lægger indhold bag bjælkerne, mens de vises. Den dynamiske tager højde for, om bjælkerne er fremme eller trukket tilbage, og specifikationen advarer selv om, at den kan ændre indholdets størrelse, mens læseren ruller, så alt nedenunder flytter sig.

Skærme har også hjørner. En telefon med runde hjørner eller et udskåret kamerahul er ikke et rektangel, og med standardværdien `auto` forbliver hele siden synlig på den. Tilføjer siden `viewport-fit=cover`, beder den om, at

viewporten bliver det rektangel, der kan tegnes uden om hele den fysiske skærm, og påtager sig selv at holde teksten fri af udskæringen med variablerne `env(safe-area-inset-*)`. De skal være nul på en rektangulær skærm, så én regel for polstringen dækker begge slags.

Fast positionering hænger også på viewporten. En fast boks placeres efter layout-viewporten, som CSS Positioned Layout giver den dynamiske viewports størrelse, så cookiebjælker, modale baggrunde og fuldskærmsmenuer står stille, mens siden ruller. Hver af dem går ud fra, at ingen forfader har overtaget rollen, og `transform`, `will-change` og `contain` kan netop gøre det.

Udskrift er den viewport, et stylesheet skrevet til skærm aldrig møder. På papir måler viewport-enhederne sideområdet, en forespørgsel på `width` spørger til sideboksen, og et fast element gentages på hver side.

Sådan ser det ud i praksis

Siderne på dette websted erklærer `width=device-width, initial-scale=1, viewport-fit=cover`, så de lægges ud i telefonens bredde og går helt ud til kanten, også under en udskæring. Stylesheetet betaler for den sidste del: sidepolstringen er `max(1rem, env(safe-area-inset-left))` og den tilsvarende regel for højre side, så teksten holder en rem fra kanten og kun rykker længere ind, hvor der sidder en udskæring. På lave skærme i liggende format hæves gulvet til `max(2rem, 7vw, env(safe-area-inset-left))`, fordi indrykningerne er nul på en telefon uden udskæring, og én rem sad for tæt på kanten.

Fuldskærmssektioner henter deres højde fra en fælles variabel sat til `100dvh` med `100vh` som reserve, og kommentaren ved siden af siger hvorfor: `100vh` kan være højere end det synlige, mens adresselinjen er fremme, og det havde ladet den næste sektion stikke op under topsektionen. To elementer går med vilje uden om variablen. Mærket i sidehovedet er dimensioneret i statisk `vh`, fordi et mærke bundet til den dynamiske viewport ville skifte størrelse, når browserens bjælker gled væk, og flytte rundt på topsektionen midt i en rulning. Den faste baggrund på berøringsskærme er `100lvh` høj, så den aldrig skifter størrelse, når adresselinjen kommer og går, og den ligger på sit eget kompositionslag, fordi berøringsbrowsere rev den op i lyse og mørke bånd under inertirulning.

I et tidligere forsidelayout havde hver serviceblok en minimumshøjde på `100dvh`, så den fyldte én skærm, så længe indholdet var lavere end skærmen. Kommentarerne i stylesheetet, gennemgået i november 2026, beskriver en test ved 852 gange 393 pixel: en bloks indhold fyldte 412, og blokkene stødte lige op ad hinanden. I det layout kostede polstring inden for de `100dvh`, via `box-sizing: border-box`, intet på en høj skærm og åbnede kun et mellemrum, når indholdet løb over.

På det dokumenterede implementeringsstadium for kvalitetsporten i august 2026 fulgte testene medieforespørgslerne. Sitets browserdrevne kvalitetsport udvidede sine tjek til en fælles tabel med toogtyve vinduesformer, hentet fra hver eneste medieforespørgsel i stylesheetet, og det trak layouttjekket alene fra 95 sekunder op til 405.

Udskrift afslørede forskellen mellem motorerne. Udskriftsstylesheetet beskriver, at én motor regner viewport-enheder ud fra sideboksen og en anden ud fra vinduet, så en side udskrevet fra et bredt vindue mistede en tredjedel af hver linje i den anden. Sidehovedets eget stylesheet beskriver medieforespørgslernes side af samme kløft: udskriften beholdt vinduets orientering, højde og bredde, så ark udskrevet fra et bredt og et smalt vindue fik sidehoveder, der afveg 47 pixel fra hinanden, indtil `screen and` forrest i forespørgslen begrænsede reglen til vinduer.

Fælden med den indeholdende blok er også dokumenteret. Arbejdet med reduceret bevægelse viste, at fem forskellige egenskaber hver især gør et element til den indeholdende blok for alt, der placeres i forhold til viewporten, og at det i stilhed havde krympet et lag, man klikker på for at lukke, til en brøkdel af skærmen.

Hvor det går galt

Telefonen viser en lillebitte skrivebordsside. Uden viewport-`meta`-elementet lægger browseren siden ud i en skrivebordsbredde og skalerer den ned, og medieforespørgslerne ser den bredde. Tilføj `width=device-width, initial-scale=1`, og lad zoom være slået til: `user-scalable=no` fjerner læserens egen måde at forstørre lille tekst på.

En fuldskærmssektion går ud over skærmens bund. `100vh` er den store viewport, så mens adresselinjen er fremme, ligger bunden af sektionen bag den, sammen med enhver knap, der står dér. Brug `svh` til at holde et element fri af

browserbjælker, der er foldet ud, og `dvh` kun dér, hvor en størrelsesændring midt i en rulning er acceptabel.

Et fuldskærmslag skrumper til en boks inde på siden. Et overordnet element med `transform`, med en `will-change`, der nævner den, eller med bestemte containment-indstillinger kan blive den indeholdende blok for `position: fixed`, og laget tager elementets størrelse. Læg den slags lag uden for ethvert transformeret element.

En udskrevet side mister slutningen af hver linje. Viewport-enheder og forespørgsler på vinduets mål siver ind i udskriften, og en motor, der regner dem ud fra vinduet, lægger arket ud til en skærm. Hold viewport-enheder ude af alt, der skal udskrives, og sæt `screen and` foran forespørgsler, der er ment for vinduer.

Læs mere

- CSS Viewport Module Level 1 — viewport-`meta`-elementet og de tre adfærdsformer for `interactive-widget`.
- CSS Values and Units Level 4: viewport-percentage lengths — den lille, den store og den dynamiske viewportstørrelse og alle de enheder, der bygger på dem.
- CSS Positioned Layout Level 3 — fast positionering, de egenskaber, der fanger den, og hvad den gør på papir.
- CSS Environment Variables Level 1 — safe-area-indrykningerne og reglen, der gør dem til nul på en rektangulær skærm.
- Media Queries Level 4 — `width`, `height` og `orientation`, målt på viewporten på skærm og på sideboksen ved udskrift.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en viewport?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/viewport/