Engineer

Responsivt design

et layout, der tilpasser sig den skærm, det læses på

Hvad det er

Responsivt design er den praksis at bygge én side, hvis layout tilpasser sig den skærm, den læses på, fra en smal telefon holdt på højkant til en bred skærm på et skrivebord, i stedet for at bygge et separat site til hver slags enhed. Den længere form, responsivt webdesign, svarer til titlen på den artikel, der gav fænomenet navn: Ethan Marcottes Responsive Web Design, udgivet 25. maj 2010 i A List Apart. Han hentede idéen fra det, han beskrev som en spirende disciplin kaldet responsiv arkitektur, hvor vægge og installationer bøjer, giver efter og udvider sig, når mennesker nærmer sig dem.

Marcotte kaldte flydende grids, fleksible billeder og media queries »the three technical ingredients for responsive web design«. Et flydende grid giver kolonnerne bredder i forhold til den plads, der er til rådighed, i stedet for i faste pixels, så layoutet strækker sig og trækker sig sammen med vinduet. Et fleksibelt billede får aldrig lov at blive bredere end den kolonne, det står i. En media query lader et stylesheet skifte mening de steder, hvor forholdstal alene holder op med at virke: når tre kolonner bliver for smalle til at læse, skifter siden for eksempel til én.

Specifikationen Media Queries Level 4, et W3C Candidate Recommendation Draft dateret 19. februar 2026, definerer en media query som et logisk udtryk, der enten er sandt eller falsk, en test af et eller andet ved den browser eller enhed, der viser dokumentet. Den test, responsive layout læner sig op ad, er `width`, som for en skærm er viewportens bredde, rullepanelet medregnet. Længder i en query er de samme CSS-længder som alle andre steder, så en pixel er her en CSS-pixel, ikke en fysisk, og `em` måles ud fra browserens oprindelige skriftstørrelse, som læseren selv kan ændre, aldrig ud fra en størrelse, siden selv sætter.

Intet af dette virker på en telefon, medmindre siden beder om det. Uden et viewport-metaelement lægger mobilbrowsere siden op, som om skærmen var et desktopvindue, typisk 980 til 1024 CSS-pixels bredt, og skalerer så resultatet ned, så alle media queries ser en desktop. CSS Viewport-udkastet kalder det en afhjælpning for sider, der kun er lavet til store skærme, og unødvendig for sider, der fungerer godt på små, og linjen `width=device-width, initial-scale=1` i eksemplet nedenfor er den måde, en side siger, at den er en af dem.

Brug begrebet i en beslutning

Brug »Hvad er responsivt design?« 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

Marcottes argument handlede lige så meget om vedligeholdelse som om udseende. I stedet for at skræddersy adskilte designs til et stadigt voksende antal enheder foreslog han at behandle dem som facetter af den samme oplevelse. Én side, én adresse og ét stylesheet betyder, at en rettelse når alle læsere på én gang, og at en ny skærmstørrelse ikke kræver et nyt site, fordi layoutet aldrig blev skrevet til en bestemt størrelse.

Zoom er den grund, mange teams undervurderer. Når man zoomer ind, bliver CSS-pixlen større, så viewporten, målt i CSS-pixels, bliver smallere. Web Content Accessibility Guidelines 2.2, en W3C-anbefaling dateret 12. december 2024, kræver på niveau AA i succeskriterium 1.4.10 Reflow, at indhold kan bruges uden at rulle i to retninger ved en bredde svarende til 320 CSS-pixels, og retningslinjerne skriver selv, at det er, hvad et vindue på 1280 CSS-pixels bliver til ved 400 procent zoom. Med den startbredde giver en læsers zoom til 400 procent i browseren på en bærbar en viewport på 320 pixels for alle sidens media queries. W3C's forklaring af kriteriet bemærker, at websites ofte bruger responsivt design til at passe til mindre viewports, og at indhold, der passer til en lille viewport, hjælper folk, der zoomer ind på større enheder, ikke kun folk med telefoner.

Bytes er den tredje grund. HTML-standarden nævner tre forskellige grunde til at tilbyde flere filer til ét billede: valg efter viewport, art direction og valg efter billedformat. Et fotografi, der fylder bredden på en telefon, kræver langt færre pixels end det samme fotografi i hele bredden på en stor skærm, og med `srcset` og `sizes` kan browseren hente den mindre fil uden noget script.

Bredden er ikke det eneste, en side kan reagere på. Media Queries Level 4 beskriver også det primære pegeredskab: `(pointer: coarse)` matcher et pegeredskab med begrænset præcision, for eksempel en berøringsskærm, og `(hover: none)` matcher et primært pegeredskab, der ikke kan holdes hen over et element, eller slet intet pegeredskab. Marcottes artikel foreslog allerede at bruge media queries til at give links større mål på små skærme.

Sådan ser det ud i praksis

En liste af kort, der viser én kolonne på en telefon, to på en tablet på tværs og tre på en bred skærm, begynder med viewport-linjen og et billede, browseren selv kan vælge:

```
<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="En tidevandspøl ved lavvande"
/>
```

Stylesheetet bærer layoutet:

```
.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;
}
```

Gridet er den flydende del: kolonner på `1fr` deler den bredde, der er, så intet har en fast størrelse. De to media queries er kontakterne, og fordi de er skrevet i `em`, følger de læserens standardskriftstørrelse. En læser, der har valgt en større standardskrift, får den ene kolonne ved vinduesbredder, hvor andre allerede ser to, og det er netop, hvad de længere linjer ville kræve. Den sidste regel holder hvert billede inde i sin kolonne.

Attributten `sizes` fortæller browseren, før den har lagt noget op, hvor bredt billedet vil blive vist: den første betingelse, der er sand, vinder, og den sidste værdi gælder, når ingen er det. Ved en viewport på 39em er den første betingelse sand, så billedet regnes for 39em bredt; ved 41em er det den anden, og billedet regnes for halvdelen af viewporten, 20,5em. Det lidt bredere vindue har brug for den mindre fil, fordi layoutet har ændret sig, og HTML-standarden gennemgår den samme effekt i sit eget eksempel. Ud fra de tal og `w`-angivelserne i `srcset` regner browseren ud, hvilken pixeltæthed hver fil ville give på denne skærm, og den må vælge en hvilken som helst af dem ud fra skærmens pixeltæthed, zoomniveauet og muligvis netværket. Attributterne `width` og `height` lader den reservere plads til billedet, før filen er hentet.

En porteføljeside om at gøre sitets browsertjek hurtigere viser den samme idé omsat til en test. I arbejdet, som blev gjort i august 2026, fik de tjek, der styrer en headless browser, én fælles viewport-tabel med toogtyve vinduesformer, udledt af alle de media queries, stylesheetet reelt indeholder, og layouttjekket, som ser på siden ved alle de vinduesformer, designet tegner, gik derfor fra 95 sekunder til 405. Formerne kommer fra stylesheetet, ikke fra en liste over enheder, så det er de breakpoints, designet faktisk har, siderne bliver tjekket ved.

Hvor det går galt

Breakpoints, der spørger til enheden. Marcottes eksempler fra 2010 brugte `max-device-width`, bredden af hele skærmen. Media Queries Level 4 erklærer nu `device-width`, `device-height` og `device-aspect-ratio` forældede, siger, at forfattere ikke må bruge dem, og bemærker, at de ofte bruges som stedfortræder for at genkende mobile enheder, selv om de intet siger om, hvor meget plads siden faktisk har. Spørg i stedet på `width`, og sæt hvert breakpoint, hvor indholdet holder op med at fungere, ikke hvor en populær telefon tilfældigvis slutter.

Et hul mellem to breakpoints. Et stylesheet med `(max-width: 320px)` til ét layout og `(min-width: 321px)` til det næste ser komplet ud, men specifikationen påpeger, at en viewport mellem 320 og 321 pixels bred matcher ingen af dem, og brøkdele af pixels forekommer: de kommer fra skærme med pixeltætheder, der ikke er hele tal, og fra zoom. Ved de bredder gælder ingen af de to blokke. Brug intervalsyntaksen, `(width <= 320px)` og `(width > 320px)`, så hver bredde falder i præcis én af dem.

Zoom slået fra. En viewport-linje med `user-scalable=no` eller med en `maximum-scale` under 2 dumper i W3C's testregel for succeskriterium 1.4.4 Resize Text, som kræver, at tekst kan forstørres til 200 procent. Reglen bemærker, at desktopbrowsere ignorerer elementet, og at de fleste moderne mobilbrowsere enten ignorerer det som standard eller tilbyder en tilgængelighedsindstilling, der alligevel tillader zoom, så kun læsere med ældre mobilbrowsere stadig rammes. Udelad begge egenskaber, og få i stedet layoutet til at tåle zoom. En mindre skrift i et smalt breakpoint kræver samme omhu: W3C's forklaring af Reflow tillader det, men kun hvis læseren stadig på en eller anden måde kan få teksten op på det dobbelte af den normale størrelse.

Billeder, der ignorerer layoutet. Med `w`-angivelser i `srcset` og ingen `sizes` lader HTML-standarden browseren gå ud fra, at billedet fylder hele viewporten, så et billede, der vises i en tredjedel af skærmen, kan hente en fil omkring tre gange bredere end nødvendigt. Procenter er ikke tilladt i `sizes`; brug `vw`. Og når `srcset` bruger `w`-angivelser, ignorerer browsere, der forstår `srcset`, attributten `src`, så en ændring af `src` alene ændrer intet for de fleste læsere. Skriv `sizes`, så den svarer til CSS'en, og ret begge dele samtidig, når layoutet ændres. At vælge billeder med et script er ingen løsning: standarden advarer om, at billedet så kan blive hentet to gange eller slet ikke for læsere, der har slået scripts fra.

Ét bredt element får hele siden til at rulle vandret. En datatabel, en lang kodelinje eller en indlejring med fast bredde kan være bredere end 320 CSS-pixels og få hele siden til at rulle vandret, så hvert afsnit skal læses ved at rulle frem og tilbage. W3C's forklaring af Reflow undtager indhold, der kræver to dimensioner, for eksempel datatabeller, kort og video, men ikke afsnittene omkring det. Placer tabellen i sin egen beholder, der ruller for sig selv med `overflow-x: auto`, og giv billeder og video `max-width: 100%`, så kun den del, der har brug for det, ruller.

At tage en smal skærm for en berøringsskærm. En bærbar, der er zoomet til 400 procent, er smal og har en mus; en stor tablet er bred og har slet ingen hover. Layout hører til `width`, mens målstørrelser og hovereffekter hører til `pointer` og `hover`. Media Queries Level 4 råder forfattere til også at overveje `any-pointer` og `any-hover`, fordi en enhed kan have mere end én måde at pege på.

Læs mere

- Responsive Web Design — Ethan Marcottes artikel fra 25. maj 2010, der gav tilgangen navn og beskrev dens tre ingredienser.
- Media Queries Level 4 — `width`-egenskaben, intervalsyntaksen, relative enheder, pointer og hover, og hvorfor `device-width` er forældet.
- HTML Standard: Images — `srcset`, `sizes`, de tre grunde til flere filer og det gennemregnede eksempel med 29em og 32em.
- Understanding Success Criterion 1.4.10: Reflow — hvorfor 320 CSS-pixels, hvordan zoom hænger sammen med det, og hvilket indhold der er undtaget.
- Meta viewport allows for zoom — W3C's testregel for `user-scalable` og `maximum-scale` med dens note om browserunderstøttelse.
- CSS Viewport Module Level 1 — viewport-metaelementet, dets egenskaber, og hvorfor mobilbrowsere går ud fra en desktopbredde.

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 responsivt design?« 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/responsive-design/