Engineer

Responsiv design

🖳 et oppsett som tilpasser seg skjermen det leses på

Hva det er

Responsiv design er praksisen med å bygge én side der oppsettet tilpasser seg skjermen den leses på, fra en smal telefon holdt på høykant til en bred skjerm på et skrivebord, i stedet for å bygge et eget nettsted for hver type enhet. Den lengre formen, responsiv webdesign, svarer til tittelen på artikkelen som ga fenomenet navn: Ethan Marcottes Responsive Web Design, publisert 25. mai 2010 i A List Apart. Han hentet ideen fra det han beskrev som en framvoksende disiplin kalt responsiv arkitektur, der vegger og installasjoner bøyer seg, gir etter og utvider seg når folk nærmer seg dem.

Marcotte kalte flytende grid, fleksible bilder og media queries «the three technical ingredients for responsive web design». Et flytende grid gir kolonnene bredder i forhold til plassen som er tilgjengelig, i stedet for i faste piksler, slik at oppsettet strekker seg og trekker seg sammen med vinduet. Et fleksibelt bilde får aldri bli bredere enn kolonnen det står i. En media query lar et stilark ombestemme seg der forholdstall alene slutter å virke: når tre kolonner blir for smale til å lese, går siden for eksempel over til én.

Spesifikasjonen Media Queries Level 4, et W3C Candidate Recommendation Draft datert 19. februar 2026, definerer en media query som et logisk uttrykk som enten er sant eller usant, en test av en eller annen side ved nettleseren eller enheten som viser dokumentet. Testen responsive oppsett lener seg på, er `width`, som for en skjerm er bredden på viewporten, rullefeltet medregnet. Lengder i en query er de samme CSS-lengdene som overalt ellers, så en piksel er her en CSS-piksel, ikke en fysisk, og `em` måles fra nettleserens opprinnelige skriftstørrelse, som leseren selv kan endre, aldri fra en størrelse siden selv setter.

Ingenting av dette virker på en telefon med mindre siden ber om det. Uten et viewport-metaelement legger mobilnettlesere opp siden som om skjermen var et skrivebordsvindu, typisk 980 til 1024 CSS-piksler bredt, og skalerer så resultatet ned, slik at alle media queries ser en stor skjerm. CSS Viewport-utkastet kaller det et tiltak for sider som bare er laget for store skjermer, unødvendig for sider som fungerer godt på små, og linjen `width=device-width, initial-scale=1` i eksempelet nedenfor er måten en side sier at den er en av dem på.

Bruk begrepet i en beslutning

Bruk «Hva er responsiv design?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Marcottes argument handlet like mye om vedlikehold som om utseende. I stedet for å skreddersy frakoblede design til et stadig voksende antall enheter foreslo han å behandle dem som fasetter av den samme opplevelsen. Én side, én adresse og ett stilark betyr at en rettelse når alle lesere samtidig, og at en ny skjermstørrelse ikke krever et nytt nettsted, fordi oppsettet aldri ble skrevet for en bestemt størrelse.

Zoom er grunnen mange team undervurderer. Når man zoomer inn, blir CSS-pikselen større, så viewporten, målt i CSS-piksler, blir smalere. Web Content Accessibility Guidelines 2.2, en W3C-anbefaling datert 12. desember 2024, krever på nivå AA i suksesskriterium 1.4.10 Reflow at innhold kan brukes uten rulling i to retninger ved en bredde tilsvarende 320 CSS-piksler, og retningslinjene sier selv at det er dette et vindu på 1280 CSS-piksler blir ved 400 prosent zoom. Med den startbredden gir en lesers zoom til 400 prosent i nettleseren på en bærbar en viewport på 320 piksler for alle sidens media queries. W3Cs forklaring av kriteriet påpeker at nettsteder ofte bruker responsiv design for å passe til mindre viewporter, og at innhold som passer i en liten viewport, hjelper folk som zoomer inn på større enheter, ikke bare folk med telefoner.

Datamengden er den tredje grunnen. HTML-standarden nevner tre ulike grunner til å tilby flere filer for ett bilde: valg etter viewport, art direction og valg etter bildeformat. Et fotografi som fyller bredden på en telefon, trenger langt færre piksler enn det samme fotografiet over hele bredden på en stor skjerm, og med `srcset` og `sizes` kan nettleseren hente den mindre filen uten noe skript.

Bredden er ikke det eneste en side kan reagere på. Media Queries Level 4 beskriver også den primære pekeenheten: `(pointer: coarse)` treffer en pekeenhet med begrenset presisjon, for eksempel en berøringsskjerm, og `(hover: none)` treffer en primær pekeenhet som ikke kan holdes over et element, eller ingen pekeenhet i det hele tatt. Marcottes artikkel foreslo allerede å bruke media queries til å gi lenker større mål å treffe på små skjermer.

Slik ser det ut i praksis

En liste med kort som viser én kolonne på en telefon, to på et nettbrett på tvers og tre på en bred skjerm, begynner med viewport-linjen og et bilde nettleseren selv kan velge:

```
<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 fjærepytt ved lavvann"
/>
```

Stilarket bærer oppsettet:

```
.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 flytende delen: kolonner på `1fr` deler den bredden som finnes, så ingenting har en fast størrelse. De to media queries er bryterne, og fordi de er skrevet i `em`, følger de leserens standard skriftstørrelse. En leser som har valgt en større standardskrift, får den ene kolonnen ved vindusbredder der andre allerede ser to, og det er nettopp det de lengre linjene ville trenge. Den siste regelen holder hvert bilde inne i kolonnen sin.

Attributtet `sizes` forteller nettleseren, før den har lagt opp noe, hvor bredt bildet skal vises: den første betingelsen som er sann, vinner, og den siste verdien gjelder når ingen er det. Ved en viewport på 39em er den første betingelsen sann, så bildet regnes som 39em bredt; ved 41em er det den andre, og bildet regnes som halve viewporten, 20,5em. Det litt bredere vinduet trenger den mindre filen, fordi oppsettet har endret seg, og HTML-standarden går gjennom den samme effekten i sitt eget eksempel. Ut fra disse tallene og `w`-angivelsene i `srcset` regner nettleseren ut hvilken pikseltetthet hver fil ville gi på denne skjermen, og den kan velge hvilken som helst av dem ut fra skjermens pikseltetthet, zoomnivået og muligens nettverket. Attributtene `width` og `height` lar den sette av plass til bildet før filen er lastet ned.

En porteføljeside om å gjøre nettstedets nettlesersjekker raskere viser den samme ideen omsatt til en test. I arbeidet fra august 2026 fikk sjekkene som styrer en hodeløs nettleser, én felles viewport-tabell med tjueto vindusformer, utledet fra hver media query stilarket faktisk inneholder, og layoutsjekken, som ser på siden ved hver vindusform designet tegner opp, gikk derfor fra 95 sekunder til 405. Formene kommer fra stilarket, ikke fra en liste over enheter, så det er de bruddpunktene designet faktisk har, sidene blir sjekket ved.

Hvor det går galt

Bruddpunkter som spør etter enheten. Marcottes eksempler fra 2010 brukte `max-device-width`, bredden på hele skjermen. Media Queries Level 4 erklærer nå `device-width`, `device-height` og `device-aspect-ratio` foreldet, sier at forfattere ikke må bruke dem, og påpeker at de ofte brukes som en stedfortreder for å oppdage mobile enheter, selv om de ikke sier noe om hvor mye plass siden faktisk har. Spør heller etter `width`, og legg hvert bruddpunkt der innholdet slutter å fungere, ikke der en populær telefon tilfeldigvis slutter.

Et hull mellom to bruddpunkter. Et stilark med `(max-width: 320px)` for ett oppsett og `(min-width: 321px)` for det neste ser komplett ut, men spesifikasjonen påpeker at en viewport mellom 320 og 321 piksler bred treffer ingen av dem, og brøkdeler av piksler forekommer: de kommer fra skjermer med pikseltettheter som ikke er hele tall, og fra zoom. Ved de breddene gjelder ingen av de to blokkene. Bruk intervallsyntaksen, `(width <= 320px)` og `(width > 320px)`, slik at hver bredde havner i nøyaktig én av dem.

Zoom slått av. En viewport-linje med `user-scalable=no` eller med en `maximum-scale` under 2 stryker i W3Cs testregel for suksesskriterium 1.4.4 Resize Text, som krever at tekst kan forstørres til 200 prosent. Regelen påpeker at skrivebordsnettlesere ignorerer elementet, og at de fleste moderne mobilnettlesere enten ignorerer det som standard eller tilbyr en tilgjengelighetsinnstilling som tillater zoom likevel, slik at bare lesere med eldre mobilnettlesere fortsatt rammes. Utelat begge egenskapene, og få heller oppsettet til å tåle zoom. En mindre skrift i et smalt bruddpunkt krever samme omtanke: W3Cs forklaring av Reflow tillater det, men bare hvis leseren fortsatt på en eller annen måte kan få teksten opp i dobbel normal størrelse.

Bilder som overser oppsettet. Med `w`-angivelser i `srcset` og ingen `sizes` lar HTML-standarden nettleseren gå ut fra at bildet fyller hele viewporten, så et bilde som vises i en tredjedel av skjermen, kan laste ned en fil omtrent tre ganger bredere enn nødvendig. Prosenter er ikke tillatt i `sizes`; bruk `vw`. Og når `srcset` bruker `w`-angivelser, ignorerer nettlesere som forstår `srcset`, attributtet `src`, så en endring av bare `src` endrer ingenting for de fleste lesere. Skriv `sizes` slik at det stemmer med CSS-en, og endre begge samtidig når oppsettet endres. Å velge bilder med et skript er ingen løsning: standarden advarer om at bildet da kan bli lastet ned to ganger, eller ikke i det hele tatt for lesere som har slått av skript.

Ett bredt element får hele siden til å rulle sidelengs. En datatabell, en lang kodelinje eller en innbygging med fast bredde kan være bredere enn 320 CSS-piksler og få hele siden til å rulle sidelengs, slik at hvert avsnitt må leses ved å rulle fram og tilbake. W3Cs forklaring av Reflow unntar innhold som trenger to dimensjoner, for eksempel datatabeller, kart og video, men ikke avsnittene rundt det. Legg tabellen i en egen beholder som ruller for seg selv med `overflow-x: auto`, og gi bilder og video `max-width: 100%`, slik at bare den delen som trenger det, ruller.

Å ta en smal skjerm for en berøringsskjerm. En bærbar som er zoomet til 400 prosent, er smal og har en mus; et stort nettbrett er bredt og har ingen hover i det hele tatt. Oppsettet hører til `width`, mens målstørrelser og hovereffekter hører til `pointer` og `hover`. Media Queries Level 4 råder forfattere til også å vurdere `any-pointer` og `any-hover`, fordi en enhet kan ha mer enn én måte å peke på.

Les mer

- Responsive Web Design — Ethan Marcottes artikkel fra 25. mai 2010, som ga tilnærmingen navn og beskrev de tre ingrediensene.
- Media Queries Level 4 — `width`-egenskapen, intervallsyntaksen, relative enheter, pointer og hover, og hvorfor `device-width` er foreldet.
- HTML Standard: Images — `srcset`, `sizes`, de tre grunnene til flere filer og det gjennomregnede eksempelet med 29em og 32em.
- Understanding Success Criterion 1.4.10: Reflow — hvorfor 320 CSS-piksler, hvordan zoom henger sammen med det, og hvilket innhold som er unntatt.
- Meta viewport allows for zoom — W3Cs testregel for `user-scalable` og `maximum-scale`, med merknaden om nettleserstøtte.
- CSS Viewport Module Level 1 — viewport-metaelementet, egenskapene, og hvorfor mobilnettlesere går ut fra en skrivebordsbredde.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er responsiv design?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/responsive-design/