Engineer

Responsiv design

en layout som anpassar sig efter skärmen den läses på

Vad det är

Responsiv design är att bygga en enda sida vars layout anpassar sig efter skärmen den läses på, från en smal telefon som hålls på högkant till en bred skärm på ett skrivbord, i stället för att bygga en separat webbplats för varje sorts enhet. Den längre formen, responsiv webbdesign, motsvarar titeln på artikeln som gav företeelsen dess namn: Ethan Marcottes Responsive Web Design, publicerad den 25 maj 2010 i A List Apart. Han hämtade idén från det han beskrev som en framväxande disciplin kallad responsiv arkitektur, där väggar och installationer böjer sig, ger efter och vidgas när människor närmar sig dem.

Marcotte kallade flytande rutnät, flexibla bilder och media queries ”the three technical ingredients for responsive web design”. Ett flytande rutnät ger kolumnerna bredder i proportion till det tillgängliga utrymmet i stället för i fasta pixlar, så att layouten sträcks ut och dras ihop med fönstret. En flexibel bild får aldrig bli bredare än kolumnen den står i. En media query låter en stilmall ändra sig där proportionerna ensamma slutar fungera: när tre kolumner blir för smala för att läsa går sidan till exempel över till en.

Specifikationen Media Queries Level 4, ett W3C Candidate Recommendation Draft daterat den 19 februari 2026, definierar en media query som ett logiskt uttryck som antingen är sant eller falskt, ett test av någon egenskap hos webbläsaren eller enheten som visar dokumentet. Testet som responsiva layouter lutar sig mot är `width`, som för en skärm är viewportens bredd, rullningslisten inräknad. Längder i en query är samma CSS-längder som överallt annars, så en pixel är här en CSS-pixel, inte en fysisk, och `em` mäts från webbläsarens ursprungliga teckenstorlek, som läsaren själv kan ändra, aldrig från en storlek som sidan själv sätter.

Inget av detta fungerar på en telefon om inte sidan ber om det. Utan ett viewport-metaelement lägger mobilwebbläsare ut sidan som om skärmen vore ett skrivbordsfönster, vanligtvis 980 till 1024 CSS-pixlar brett, och skalar sedan ner resultatet, så att alla media queries ser en stor skärm. CSS Viewport-utkastet kallar det en lindring för sidor som bara är gjorda för stora skärmar, onödig för sidor som fungerar bra på små, och raden `width=device-width, initial-scale=1` i exemplet nedan är sättet en sida säger att den är en av dem.

Använd begreppet i ett beslut

Använd ”Vad är responsiv design?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Styrde full stack-utvecklingen av GIS-kartor — kartutveckling i full stack, från databas till webbläsare

Styrde full stack-utvecklingen av GIS-kartor Hör av er →

Varför det spelar roll

Marcottes argument handlade lika mycket om underhåll som om utseende. I stället för att skräddarsy fristående designer för ett ständigt växande antal enheter föreslog han att man skulle behandla dem som fasetter av samma upplevelse. En sida, en adress och en stilmall innebär att en rättelse når alla läsare på en gång, och att en ny skärmstorlek inte kräver en ny webbplats, eftersom layouten aldrig skrevs för en viss storlek.

Zoom är skälet som många team underskattar. När man zoomar in blir CSS-pixeln större, så viewporten, räknad i CSS-pixlar, blir smalare. Web Content Accessibility Guidelines 2.2, en W3C-rekommendation daterad den 12 december 2024, kräver på nivå AA i framgångskriterium 1.4.10 Reflow att innehåll kan användas utan rullning i två riktningar vid en bredd som motsvarar 320 CSS-pixlar, och riktlinjerna skriver själva att det är vad ett fönster på 1280 CSS-pixlar blir vid 400 procents zoom. Med den utgångsbredden ger en läsares zoom till 400 procent i webbläsaren på en bärbar dator en viewport på 320 pixlar för varje media query på sidan. W3C:s förklaring av kriteriet konstaterar att webbplatser ofta använder responsiv design för att passa mindre viewporter, och att innehåll som ryms i en liten viewport hjälper personer som zoomar in på större enheter, inte bara personer med telefoner.

Datamängden är det tredje skälet. HTML-standarden nämner tre olika skäl att erbjuda flera filer för en bild: val efter viewport, art direction och val efter bildformat. Ett fotografi som fyller bredden på en telefon behöver långt färre pixlar än samma fotografi över hela bredden på en stor skärm, och med `srcset` och `sizes` kan webbläsaren hämta den mindre filen utan något skript.

Bredden är inte det enda en sida kan svara på. Media Queries Level 4 beskriver också det primära pekdonet: `(pointer: coarse)` matchar ett pekdon med begränsad precision, till exempel en pekskärm, och `(hover: none)` matchar ett primärt pekdon som inte kan hovra över ett element, eller inget pekdon alls. Marcottes artikel föreslog redan att media queries kunde ge länkar större ytor att träffa på små skärmar.

Så ser det ut i praktiken

En lista med kort som visar en kolumn på en telefon, två på en surfplatta på tvären och tre på en bred skärm börjar med viewport-raden och en bild som webbläsaren själv kan välja:

```
<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 tidvattenspöl vid lågvatten"
/>
```

Stilmallen bär layouten:

```
.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;
}
```

Rutnätet är den flytande delen: kolumner på `1fr` delar på den bredd som finns, så ingenting har en fast storlek. De två media queries är omkopplarna, och eftersom de är skrivna i `em` följer de läsarens standardteckenstorlek. En läsare som har valt en större standardstorlek får den enda kolumnen vid fönsterbredder där andra redan ser två, och det är precis vad de längre raderna skulle behöva. Den sista regeln håller varje bild inne i sin kolumn.

Attributet `sizes` talar om för webbläsaren, innan den har lagt ut något, hur brett bilden kommer att visas: det första villkoret som är sant vinner, och det sista värdet gäller när inget är det. Vid en viewport på 39em är det första villkoret sant, så bilden räknas som 39em bred; vid 41em är det det andra, och bilden räknas som halva viewporten, 20,5em. Det lite bredare fönstret behöver den mindre filen, eftersom layouten har ändrats, och HTML-standarden går igenom samma effekt i sitt eget exempel. Utifrån de talen och `w`-angivelserna i `srcset` räknar webbläsaren ut vilken pixeltäthet varje fil skulle ge på den här skärmen, och den får välja vilken som helst av dem med hänsyn till skärmens pixeltäthet, zoomnivån och möjligen nätverket. Attributen `width` och `height` låter den reservera plats för bilden innan filen har laddats ner.

En portföljsida om att göra webbplatsens webbläsarkontroller snabbare visar samma idé omsatt i ett test. I arbetet från augusti 2026 fick kontrollerna som styr en huvudlös webbläsare en gemensam viewport-tabell med tjugotvå fönsterformer, härledd ur varje media query som stilmallen faktiskt innehåller, och layoutkontrollen, som granskar sidan vid varje fönsterform som designen är ritad för, gick därför från 95 sekunder till 405. Formerna kommer från stilmallen, inte från en lista över enheter, så det är de brytpunkter designen faktiskt har som sidorna kontrolleras vid.

Var det går fel

Brytpunkter som frågar efter enheten. Marcottes exempel från 2010 använde `max-device-width`, bredden på hela skärmen. Media Queries Level 4 förklarar nu `device-width`, `device-height` och `device-aspect-ratio` föråldrade, säger att författare inte får använda dem och påpekar att de ofta används som ombud för att upptäcka mobila enheter, trots att de inte säger något om hur mycket plats sidan faktiskt har. Fråga i stället efter `width`, och lägg varje brytpunkt där innehållet slutar fungera, inte där en populär telefon råkar ta slut.

Ett glapp mellan två brytpunkter. En stilmall med `(max-width: 320px)` för en layout och `(min-width: 321px)` för nästa ser komplett ut, men specifikationen påpekar att en viewport mellan 320 och 321 pixlar bred matchar ingen av dem, och bråkdelar av pixlar förekommer: de kommer från skärmar med pixeltätheter som inte är heltal och från zoom. Vid de bredderna gäller inget av de två blocken. Använd intervallsyntaxen, `(width <= 320px)` och `(width > 320px)`, så att varje bredd hamnar i exakt en av dem.

Zoom avstängd. En viewport-rad med `user-scalable=no` eller en `maximum-scale` under 2 underkänns av W3C:s testregel för 1.4.4 Resize Text, det framgångskriterium som kräver att text kan förstoras till 200 procent. Regeln noterar att skrivbordswebbläsare ignorerar elementet och att de flesta moderna mobilwebbläsare antingen ignorerar det som standard eller erbjuder en tillgänglighetsinställning som tillåter zoom ändå, så att bara läsare med äldre mobilwebbläsare fortfarande drabbas. Utelämna båda egenskaperna, och se i stället till att layouten tål zoom. En mindre teckenstorlek i en smal brytpunkt kräver samma omsorg: W3C:s förklaring av Reflow tillåter det, men bara om läsaren fortfarande på något sätt kan få texten till dubbel normal storlek.

Bilder som struntar i layouten. Med `w`-angivelser i `srcset` och ingen `sizes` låter HTML-standarden webbläsaren anta att bilden fyller hela viewporten, så en bild som visas i en tredjedel av skärmen kan ladda ner en fil ungefär tre gånger bredare än nödvändigt. Procent är inte tillåtet i `sizes`; använd `vw`. Och när `srcset` använder `w`-angivelser ignorerar webbläsare som förstår `srcset` attributet `src`, så en ändring av enbart `src` märks inte för de flesta läsare. Skriv `sizes` så att det stämmer med CSS:en, och ändra båda samtidigt när layouten ändras. Att välja bilder med ett skript är ingen lösning: standarden varnar för att bilden då kan laddas ner två gånger, eller inte alls för läsare som har stängt av skript.

Ett brett element får hela sidan att rulla i sidled. En datatabell, en lång kodrad eller en inbäddning med fast bredd kan vara bredare än 320 CSS-pixlar och få hela sidan att rulla i sidled, så att varje stycke måste läsas genom att rulla fram och tillbaka. W3C:s förklaring av Reflow undantar innehåll som behöver två dimensioner, till exempel datatabeller, kartor och video, men inte styckena runt omkring. Lägg tabellen i en egen behållare som rullar för sig själv med `overflow-x: auto`, och ge bilder och video `max-width: 100%`, så att bara den del som behöver det rullar.

Att ta en smal skärm för en pekskärm. En bärbar dator som är zoomad till 400 procent är smal och har en mus; en stor surfplatta är bred och kan inte hovra alls. Layouten hör till `width`, medan målstorlekar och hovereffekter hör till `pointer` och `hover`. Media Queries Level 4 råder författare att också överväga `any-pointer` och `any-hover`, eftersom en enhet kan ha mer än ett sätt att peka.

Läs mer

- Responsive Web Design — Ethan Marcottes artikel från den 25 maj 2010 som gav arbetssättet ett namn och beskrev dess tre ingredienser.
- Media Queries Level 4 — egenskapen `width`, intervallsyntaxen, relativa enheter, pointer och hover, och varför `device-width` är föråldrad.
- HTML Standard: Images — `srcset`, `sizes`, de tre skälen till flera filer och det genomräknade exemplet med 29em och 32em.
- Understanding Success Criterion 1.4.10: Reflow — varför 320 CSS-pixlar, hur zoom hänger ihop med det och vilket innehåll som undantas.
- Meta viewport allows for zoom — W3C:s testregel för `user-scalable` och `maximum-scale`, med dess not om webbläsarstöd.
- CSS Viewport Module Level 1 — viewport-metaelementet, dess egenskaper och varför mobilwebbläsare utgår från en skrivbordsbredd.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är responsiv design?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/responsive-design/