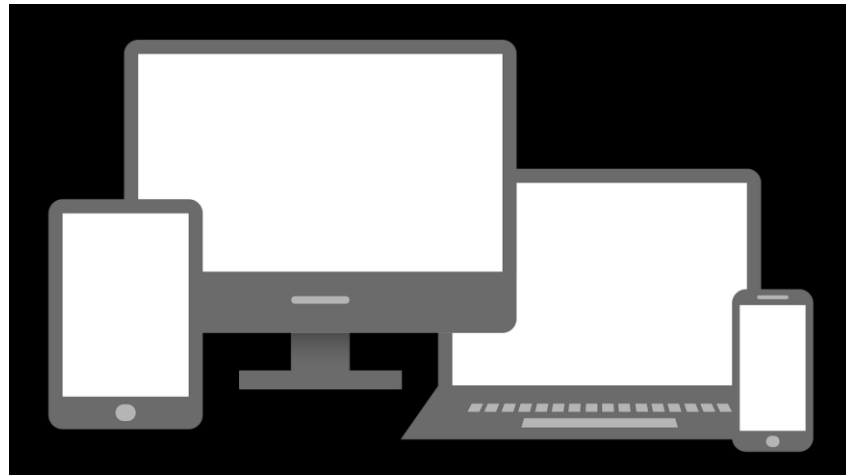


Responsiv webbdesign



Surf-statistik

- I hela världen är det fler personer som surfar från mobilen jämfört med datorn, [statcounter](#).
- I Sverige använder 85% av befolkningen internet i mobilen och 76% av befolkningen gör det dagligen, [undersökning från Internet Stiftelsen i Sverige](#).
- På svenska e-handelsplatser använder 67% av alla besökare mobilen, [undersökning från e-handel.se](#).

Mobila webben - utmaningar

- Mobiler har små skärmar, svårt att göra bra layout & design.
- Bandbredden kan vara opålitlig, kan tappa kontakten med internet ibland.
- Inmatning är inte lika lätt på en liten skärm jämfört med ett tangentbord.

Riktlinjer för en mobil webbplats

- Undvik att försöka få plats med samma mängd data på en mobil- som på en datorskärm. Det som ser bra ut på en stor skärm blir svårare att urskilja på en mobilskärm och gör att mobilgränssnittet känns svårt för användaren.
- Försök att begränsa användarinteraktion till knappar, checkboxar och alternativknappar. Det är besvärligt att fylla i textfält på en mobil.
- Texten ska vara läsbar utan att zooma.
- Placera länkar på sådant avstånd att inte användaren klicka fel av misstag.
- Innehållet ska vara åtkomligt utan att scrolla i sidled eller zooma.
- Använd inte mer och större bilder än nödvändigt. De bilder som finns ska vara komprimerade.

Mobile First

- Mobile First betyder att man i sin webbutveckling börjar med att designa och utveckla för den mobila upplevelsen först, istället för det som var vanligt förut, dvs att mobila sidor gjordes som nedskalade versioner av dator-designen.
- Mobile First bygger på två idéer:
 - Mobilt är viktigare än desktop eftersom fler surfar på mobilen.
 - Det är lättare att designa för mobilen först och skala upp än tvärtom.

Mobile-first index

- Google indexerar (=rankar) webbplatser och appar utifrån hur väl de fungerar i mobilen. Det innebär alltså att ju bättre en webbplats fungerar i mobilen desto bättre google-ranking ([SEO](#)) får sidan.

3 kategorier

- Man brukar dela in mobilappar i 3 olika kategorier

1. Native

En app utvecklad speciellt för det operativsystem som mobilen har. Native-appar installeras för Android-mobiler från Google Play och från iPhone från App Store.

+ bra prestanda eftersom är anpassad för hårdvaran på mobilen.

+ kan utnyttja all hårdvara som finns i mobilen, som kamera, mikrofon, gps etc.

- varje operativsystem/enhet (Android, iPhone) kräver en egen utvecklad app.

2. Webb

En webb-app använder webbtekniker som HTML, CSS och Javascript. Själva appen använder mobilens webbläsare.

+ den fungerar på alla typer av mobiler som har en webbläsare.

+ Med responsiv webbdesign kan samma webbsida användas både på dator, surfplatta och mobilen.

- kommer inte åt all hårdvara på mobilen.

- har ofta sämre prestanda än native-appar.

3. Hybrid

Är en kombination av webbapp och native-app.

+ - har till en viss del tillgång till hårdvarufunktionalitet.

+ - I övrigt fungerar hybridapplikationen som en webbapplikation.

Webb-appar i kursen

- Användare surfar på Internet med enheter med olika skärmstorlek, t.ex. TV:n, dataskärmar, mobiler och surfplattor.
- Hur kan webbutvecklare se till att en webbplats är snygg och läsbar på alla enheter, oavsett skärmstorlek?

Responsiv webbdesign

Responsiv webbdesign är en webbplats förmåga att ändra storlek och organisera om innehållet baserat på:

- Storleken på webbläsarfönstret.
- Storleken på annat innehåll på webbplatsen.

EXEMPEL

<http://www.bostonglobe.com>

Absolut mått

- Enheten px (pixlar), som används på webbplatser för att sätta storlekar, är ett absolut värde.
- När en skärmstorlek ändras (t.ex. byter från landscape till portrait på en mobil) kan element som har en storlek satt i pixlar visas för små, flöda över skärmen eller bli helt oläslig.



Relativa mått

- Relativa mått är bättre än absoluta mått, eftersom de tillåter att proportionerna på en webbplats är samma oavsett skärmstorlek eller layout.
- Använd relativa mått (t.ex. `em`, `rem` och `%`) för att få en responsiv webbdesign.

Kom ihåg!

- Relativa måttenheter är ett första steg mot att införa responsiv design på en webbplats.



Relativa måttet **em**

- Måttenhet em , är storleken på basfonten som används.
- Normalt är webbläsarens basfont är 16 pixlar, så en $1\ em$ är lika med 16 pixlar, $2\ em$ skulle vara 32 pixlar, och så vidare.
- Ett elements em beror på förälderns storlek på basfonten.

Relativa måttet **rem**

- Måttet `rem` jämför med samma textstorlek, normalt 16 pixlar.
- Det gör det enklare att veta hur stort eller litet texten kommer att visas.
- Beror **inte** på storleken på förälderns basfont.

em eller rem ?

- Om du sätta storleken på element över hela webbplatsen är **rem** den bästa enheten.
- Om du vill sätta storleken på ett element i jämförelse med andra element i närheten är **em** enheten som ska användas.

Relativa måttet %

- Procent (%) används vanligtvis för att sätta storleken på värden som `width`, `height`, `border`, `padding` och `margin`.
- När procent (%) används för `width`, `height`, `padding` och `margin` kommer storleken att sättas i förhållande till **förälder-elementet**.

Minsta och max-bredd

- Även om relativa mått ger responsiv design kan element på en webbplats bli felaktiga/fula när de blir för små eller stora.
- **Minsta** och **maximala** bredden på ett element kan ställas in med `min-width` och `max-width`.

Kom ihåg!

- Minsta och maximala höjden på ett element kan ställas in med `min-height` och `max-height`.



Skala bilder & videos

Många webbplatser innehåller en mängd olika media, som bilder och videoklipp.

Det är viktigt att de skalas proportionellt så att användarna kan se dem korrekt.

https://www.w3schools.com/howto/howto_css_image_responsive.asp

Skala bilder & videos

Många webbplatser innehåller en mängd olika media, som bilder och videoklipp. Det är viktigt att de skalas proportionellt så att användarna kan se den korrekt.

```
.container {  
  width: 50%;          /* Half browser width */  
  height: 200px;  
  overflow: hidden;   /* Larger content will be hidden */  
}  
  
.container img {  
  max-width: 100%;    /* 100% of parent (.container) */  
  height: auto;       /* Scales proportionally to width */  
  display: block;  
}
```

Det är värt att memorera hela exemplet ovan. Det representerar ett mycket vanligt **designmönster** som används för att skala bilder och videor proportionellt.

HTML

```
<section class="image-container">  
    
</section>
```

CSS

```
.image-container {  
  overflow: hidden;  
}  
  
.image-container img {  
  max-width: 100%;  
  height: auto;  
  display: block;  
  margin: auto;  
}
```


Kom ihåg!

- När `height` på en bild eller video är satt, kan `width` sättas till `auto`, så skalar det proportionerligt.
Eller så kan `width` sättas och sedan sätta `height` till `auto`.



Skala bakgrundsbilder

- Bakgrundsbilder kan också skalas responsivt.

```
body
{
  background-image: url('camel-background.jpg');
  background-repeat: no-repeat; /* Don't repeat */
  background-position: center; /* Center image */
  background-size: cover; /* Scale proportionally */
}
```

Bilden kommer att täcka hela bakgrunden av elementet, samtidigt som bilden hålls skalenlig. Om bildens storlek överstiger förälder-elementets storlek kommer endast en del av bilden att visas.

HTML

```
<header id="banner">  
  <h1>Bana's Travel Blog</h1>  
</header>
```

CSS

```
#banner {  
  height: 46rem;  
  background-image: url('camelbackground.jpg');  
  background-position: center;  
  background-size: cover;  
  background-repeat: no-repeat;  
}
```

Kom ihåg!

- En bakgrundsbild av ett HTML-element kommer att skala proportionellt om `background-size` sätts till `cover`.





Sammanfattning



- Relativa måttenheter är ett första steg mot att införa responsiv design på en webbplats.
- Enheten för `em` sätter textstorleken i förhållande till textstorleken för förälderelementet.
- Enheten av `rem` sätter textstorleken i förhållande till textstorleken för rotelementet `<html>`.
- Procent används vanligtvis för att sätta storleken på boxmodell-värden, som `width`, `height`, `border`, `padding` och `margin`.
- När procent används för `width` och `height`, kommer storleken att sättas i förhållande till förälderelementet.
- Procent kan användas för att ställa in `padding` och `margin`. De sätts i förhållande till bredden på förälderelementet.
- Minsta och maximala bredden på ett element kan ställas in med `min-width` och `max-width`.
- Minsta och maximala höjden på ett element kan ställas in med `min-height` och `max-height`.
- När `height` på en bild eller video är satt, kan `width` sättas till `auto`, så skalar det proportionerligt. Eller så kan `width` sättas och sedan sätta `height` till `auto`.
- En bakgrundsbild av ett HTML-element kommer att skala proportionellt om `background-size` sätts till `cover`.

Responsiv webbdesign

- Absoluta mått (t.ex. px) 
- Relativa mått (t.ex. %, rem, em) 
- Relativa mått gör att sidan skalar om utseendet, men hur får vi olika utseende beroende på skärmstorlek?

EXEMPEL

<http://www.bostonglobe.com>

Responsiv webbdesign

Använd "Media queries" i **CSS** för att skilja på olika storlekar.

```
@media screen and (max-width: 600px)
{
  /* CSS för mobiler och små skärmar */
}
@media screen and (min-width: 601px) and (max-width: 991px) {
  /* CSS för surfplattor och mellanbreda skärmar */
}
@media screen and (min-width: 992px) {
  /* CSS för datorskärmar och breda skärmar */
}
```

HTML – index.html



```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Responsiv webbdesign</h1>
</body>
</html>
```


CSS – style.css

```
@media screen and (max-width: 600px)
{
  body {
    background-color: blue;
  }
}
@media screen and (min-width: 601px) and (max-width: 991px)
{
  body {
    background-color:green;
  }
}
@media screen and (min-width: 992px)
{
  body {
    background-color:yellow;
  }
}
```

Responsiv webbdesign

2 saker krävs för att göra bra responsiv webbdesign

- Relativa mått (t.ex. %, rem, em) 
- Använd "media queries" för att ändra utseende (layout) beroende på skärmstorlek. 

CSS Grid

- CSS-Grids underlättar designen av responsiva layouter.
- relativt nytt (mars 2017).
- en metod för att bygga upp en layout med CSS.
- bra för att strukturera webbsidor med både rader och kolumner (tvådimensionella element).

CSS Grid

- ny måttenhet för bredd och höjd, nämligen fraction (fr), som är en del av helheten.
- döpa rader och kolumner, för att lättare hålla isär dem.
- skapa layouter helt oberoende av ordningen av element i HTML
- Tillsammans med media queries kan layouten se väldigt olika ut för enheter med olika skärmstorlek.

CSS Grid

CSS Grids tillsammans med **media queries** ger förutsättningar för att göra en bra responsiv webbdesign



HTML

```
<div id="content">  
  <section id="skolverket">  
  </section>  
  <section id="innehall">  
  </section>  
  <section id="begrepp">  
  </section>  
  <section id="links">  
  </section>  
  <section id="ovning">  
  </section>  
  <section id="inlupp">  
  </section>  
  <section id="fordjupning">  
  </section>  
</div>
```

CSS

```
section#skolverket { grid-area: skolverket; }
section#innehall { grid-area: innehall; }
section#begrepp { grid-area: begrepp; }
section#links { grid-area: links; }
section#ovning { grid-area: ovning; }
section#inlupp { grid-area: inlupp; }
section#fordjupning { grid-area: fordjupning; }

#content {
  display: grid;
  padding-top: 6rem;
  padding-bottom: 1rem;
  padding-right: 1rem;
  padding-left: 1rem;
  grid-template-columns: 1fr 1fr 1fr 1fr;
  grid-auto-rows: minmax(50px, auto);
  grid-gap: 1rem;
  margin: 0 auto;
  grid-template-areas:
    "innehall innehall innehall links"
    "ovning ovning begrepp begrepp"
    "inlupp inlupp fordjupning fordjupning"
    "skolverket skolverket skolverket skolverket";
}
```