

Individuell Övning

Prestandaoptimering (ca 30 min)

Lazy Loading, Bildoptimering och Throttling

Introduktion

Du får en sida som medvetet är dåligt optimerad ur prestandaperspektiv:

- stora bilder
- ingen lazy loading
- ingen bildoptimering
- en scroll-händelse som loggar alltför ofta

Din uppgift är att förbättra sidans prestanda utan att ändra innehållet.

Fokuset ligger på bildoptimering och händelsehantering för bättre upplevd prestanda.

Öppna projektet i VS Code och starta index.html med Live Server.

Er uppgift:

Steg 1

Sätt meningsfulla alt-texter

1. Öppna **index.html**.
2. Ge följande bilder korrekta och beskrivande **alt**-texter:
 - hero-bilden
 - alla produktbilder

Tänk på att en alt-text ska beskriva vad bilden föreställer, t.ex.:

“Svart kaffekopp på träbord”, “Vit stol vid skrivbord”, eller annat som passar.

Detta förbättrar tillgänglighet och UX, och gör sidan mer robust.

Steg 2

Lazy loading med `loading="lazy"`

Lägg till lazy loading på alla produktbilder:

- Hero-bilden ska **inte** ha `loading="lazy"` (den ska synas direkt).
- Alla bilder med klassen `.product-image` ska få attributet: `loading="lazy"`

Steg 3

Gör en produktbild responsiv med `srcset` och `sizes`

1. Välj **minst en** produktbild (t.ex. Produkt 1).
2. Lägg till två eller tre bildstorlekar i `srcset`.
3. Lägg till `sizes` så att mobilen får en mindre bild.
4. Behåll `loading="lazy"`.

`srcset` visar vilka bredder som finns, och `sizes` berättar hur stor bilden ungefär visas i olika skärmbredder. Webbläsaren väljer rätt bild automatiskt.

Steg 4,

Stabilisera layouten (ca 5 min)

När bilder saknar definierade mått kan sidan “hoppa runt” när de laddas.
Välj ett av följande sätt för att reservera plats:

Alternativ A, Ange `width + height` i HTML

Exempel:

```

```

Alternativ B, Använd CSS `aspect-ratio`

Lägg in i `style.css`:

```
.hero-image {  
  display: block;  
  width: 100%;  
  height: auto;  
  aspect-ratio: 16 / 6;  
}  
  
.product-image {  
  display: block;
```

```
width: 100%;  
height: auto;  
aspect-ratio: 16 / 9;  
}
```

Båda sätten gör att layouten inte hoppar när bilder laddas.

Steg 5

Kort mätning i DevTools

1. Öppna DevTools → Network
2. Ladda om sidan **innan** dina optimeringar
3. Ladda om sidan **efter** dina optimeringar
4. Jämför:
 - Hur många bilder laddas direkt?
 - Känns sidan snabbare?

Ingen skriftlig redovisning krävs, det räcker att du reflekterar själv.

Klart när du har:

- Beskrivande alt-texter på alla bilder
- Lazy loading på produktbilder
- Minst en responsiv `srcset`-bild
- Stabilitet i layouten (inga hopp)
- Testat skillnaden i Network-panelen