

# Inlämningsuppgift #2 - CSS

## Deadline

Se lärplattform

## Bakgrund

I den här inlämningsuppgiften jobbar alla med samma design.

## Frågor om designen

Karin har blivit klar med sitt designarbete och tagit semester, så det är ju just typiskt. Du får helt enkelt improvisera på de bitar som är otydliga i briefen angående designen.

## Tillvägagångssätt/arbetsordning

Ta ett djupt andetag och ta det lugnt, när du ser designen. Strukturerat och metodiskt arbete kommer att få dig klar på nolltid. Planera ditt arbete i förväg, tänk igenom och sätt därefter igång.

Vecka 1 gick vi igenom HTML från grunden, och hur du ska tänka kring taggar. Vecka 2 är det fullt fokus på CSS.

Jag skulle föreslå följande arbetsgång:

- Titta på designen, och försök visualisera HTML:en bakom designen.
- Vilka element är gemensamma för mobilen och desktop, vilka ändras?
- Har du verkligen läst igenom *hela* briefen? Strukturerar upp briefen och gör en checklista av den. Bara för att Karin skriver massor, behöver du inte skapa "overhead" för dig själv genom att inte *skriva om briefen till en checklista*.
- Börja med mobilversionen, lägg upp HTML:en.
- Därefter lägger du på lite enklare CSS, och kollar hur layouten flödar mellan mobil och desktop. Behöver du lägga till ytterligare brytpunkter? Behöver du begränsa hur stort innehållet blir?
- När du ser att "HTML-flödet" lilar med CSS:en, så kan du börja finlira med typografi, bilder, osv.
- Tips! Bryt ner designen i enklare komponenter. T.ex. om du behöver göra en bakgrundsbild som täcker hela sidan, fokusera på den som ett "enskilt moment" och se till att den fungerar, istället för att jobba lite med allt på en gång. Fastnar du, byt till en annan uppgift från din "checklista".

Men, det här är bara tips och du lägger såklart upp arbetet på ett sätt som passar dig.

## Följande moment/aspekter examineras

- Skapa navigationssystem
- Hantera en textlogotyp
- Positionering
- Bakgrundsbilder, skalning
- Media queries
- CSS-selektorer, inkl. Pseudoselektorer
- CSS-enheter, framför allt responsiva sådana
- Box model
- Layout (inkl. float)
- Webbfonter
- Optimerade bilder
- Självständigt arbete och problemlösning

## Betygskriterier

### För G ska du:

- Ha gjort en fungerande sida på mobil och desktop
- Sidan kanske inte är helt perfekt och du har missat några delar ur briefen, det är okej.
- HTML ska vara strukturerad korrekt och semantisk
- Du ska använda en extern CSS-fil
- Färger, bilder, typsnitt, etc. ska vara optimerad och inladdade på ett korrekt sätt (vad är en dekorativ bild, vad är en för stor/tung bild, behövs hela typsnittsfamiljen, osv.)
- Det finns lite skavanker här och där, både i utseende och i koden

### För VG ska du:

- Utöver kriterierna för G...
- Använda Sass
- Ha löst problemen i uppgiften på ett optimalt sätt och utan onödiga "hack".
- Se till att toningar och övergångar fungerar snyggt (du behöver inte använda JavaScript).
- Ha optimerat resurser och laddningstid på sajten.

# Lämna in

- All din kod zippad, inkl. genererad CSS om du har använt Sass
- Skärmdumpar på sidans utseende för såväl mobil, tablet som desktop i samtliga stora webbläsare (Firefox, Chrome, Safari, Edge) samt skärmdump på sidan i en mobil enhet
- Valideringsrapport (PDF)

## Verktyg

Här är lite tips på verktyg:

- Gimp är ett gratis och kraftfullt **bildbehandlingsprogram**: <https://www.gimp.org/>
- Här finns ett **color picker-verktyg** som app: <https://colorpicker.fr/>  
Vill du inte ladda ner en app, så är mitt tips att du lägger bilden lite bakom sidan och försöker hitta rätt nyans. Det finns också verktyg för att extrahera färger från en bild, t.ex. <https://labs.tineye.com/color/> (Vet inte hur bra det funkar)
- Tänk på vilka **resurser** du har! Snacka med varandra i klassen! Hur tänker du? Bilda studiegrupper och kom överens om en tid att t.ex. i 10 minuter prata om hur ni tänker kring olika moment. Kanske går ni igenom "brief-checklistan" och jämför med varandra? Vad har ni missat, vad tänkte någon annan på men inte du?
- CDN-länk (content delivery network, s k "lastbalansering" av resurser) till FontAwesome: <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css> Vill du inte ha med alla så har du specifika här: <https://cdnjs.com/libraries/font-awesome>

## Menyn

För menyn kan du göra på två sätt, antingen med ren HTML eller JavaScript. Båda är godkända och det spelar ingen roll vilken du använder, och det spelar ingen roll för G/VG. Här är ett exempel på hur "inputs" används för en CSS-only toggle menu:

<https://codepen.io/Sfate/pen/nLBGr>

Det andra alternativet är att du lägger in JavaScript i slutet på ditt HTML-dokument mellan `<script>...</script>`-taggar, här är ett exempel nedan (du hittar det som bifogad fil också i resurserna).

Det som händer är att varje gång div:en med id:t "menu-button" klickas på, så växlar den mellan att ha en klass som heter "menu-open" och inte. Använd den för att kontrollera utseendet.

Se kod på nästa sida.

```
<div id="menu-button">hamburgare</div>
<!-- din andra html-kod -->
<script>
const menuBtn = document.querySelector('#menu-button');
menuBtn.addEventListener('click', () => {
  menuBtn.classList.toggle('menu-open');
});
</script>
```