

Da jeg først begynte på eksamenen, tenkte jeg først på at C-kravet var et relativt godt utgangspunkt å gå etter. Men etter å ha kommet godt i gang, og blitt ferdig med HTMLen til C-kravet, kom jeg på ulike problemer, den største av dem var at nettsiden skulle være responsiv.

Jeg synes at responsivt nettsted er vanskelig, og dermed valgte jeg å gå ned til en D. Jeg vil heller ha en solid D enn å være usikker på å få en C.

De andre problemene jeg snublet over var, at det står egentlig ikke at nettstedet skal være responsiv i D-kravet, kun at det skal tilpasse media queries "Fra D nivået er det krav om media queries og fra C er det krav om både CSS og flex og grid".

Jeg spurte om det og fikk svar at den skal være responsiv, og det har jeg forhåpentligvis gjort.

Jeg har gått mange ganger fram og tilbake fra demoen til eksamensteksten, hvor det ser ut som det har blitt gjort på den måten, men ifølge eksamensteksten har det blitt gjort på en helt annen måte, og dermed måtte jeg endre mange seksjoner osv.

Siden jeg begynne på C-kravet, starter kildene mine der, men etter hvert som jeg har jobbet med oppgaven, har jeg egentlig bare redigert inne i inspect på FireFox, og brukt mine egne notater og pensum som kilder.

Link til Github repository:

- <https://github.com/AFKJonas/Eksamen-webutvikling-2024-Jonas-Markus-A.E>

Her er kildene jeg brukte:

Index.html

Generering av randomisert tekst:

- <https://www.lipsum.com/>

generelt:

- Bildene er lenket i html/css
- Ikoner er ifra: <https://fontawesome.com/search?p=12> (@Fonticons, Inc.)
- Validering av nettsted: <https://validator.w3.org/nu/#textarea>

Problemer jeg møtte på underveis:

- Tok hele koden og la den inn i validatoren (<https://validator.w3.org/>) og fikk warning på **Warning: Article lacks heading. Consider using h2-h6 elements to add identifying headings to all articles.**. plasserte flere headers, slik at det ikke oppsto warningene, men senere slettet de, da Warnings er lovt å ha.
- Så på eksempelbildet at det var en horizontal linje rett over footeren og søkte på "how to make a line on html" og trykket på første linken: (<https://www.geeksforgeeks.org/how-to-add-horizontal-line-in-html/>) Der fikk jeg vite om et nytt element kalt <hr> og slik fikk jeg en linje.

Slettet den senere, da jeg fant ut av at man kan markere top-border inni inspeksjonen på FireFox, og valgte istedenfor å lage en horizontal linje i css enn html.

- Fant ut av når man linket til en annen side, bruker man href="index#section" og ikke href="index.html/#"section". Måtte spør chat.gpt:

href="index.html/#"section?

Erstattet:

"href="index.html/#"section"
til
href="index#section."

The value href="index.html/#"section? seems to be a malformed or incorrectly structured URL. Here's a breakdown of the issues and how to correct them:

Understanding href:

1. href Attribute: In HTML, the href attribute is used in anchor tags <a> to specify the URL of the link's destination.

What's Wrong?

- Double Quotation Marks: The inner double quotation marks (") conflict with the outer double quotation marks, making it invalid HTML. This needs to be corrected.
- Unnecessary Hash (#): The # is typically used to link to an ID on the page (e.g., #section). Including it here as #" adds confusion.
- Query String Syntax (?): A query string starts with a ?, but in this context, it's unclear what it's supposed to do.

Possible Fixes

Here are some corrected examples based on potential intentions:

1. Linking to a Section on the Page

If you're trying to link to a section with an ID, remove the extra syntax:

```
html
<a href="index.html#section">Go to Section</a>
```

This links to an element with the id="section" on index.html.

2. Using a Query String

If you are attempting to include a query parameter:

```
html
<a href="index.html?section=1">Go to Section 1</a>
```

This could be processed by a script on index.html to load section=1.

3. Relative Links

If you only want to refer to the index.html page:

```
html
<a href="index.html">Go to Index</a>
```

4. Linking to a Placeholder

If you're using a placeholder for future navigation:

```
html
<a href="#">Placeholder Link</a>
```

Let me know if you were aiming for a specific structure!

🔊 📄 📌 🔄 🔄



Activate W

Her er også et annet spørsmål jeg spurte ChatGPT:

Kan man putte section ved siden av hverandre?



Ja, det er absolutt mulig å plassere <section>-elementer ved siden av hverandre! Dette kan du gjøre ved hjelp av CSS, spesielt ved å bruke flexbox eller grid. Her er to måter å gjøre det på:

Metode 1: Flexbox

Flexbox er en enkel og fleksibel måte å plassere elementer horisontalt.

HTML

```
html
<div class="container">
  <section class="section1">Innhold i seksjon 1</section>
  <section class="section2">Innhold i seksjon 2</section>
  <section class="section3">Innhold i seksjon 3</section>
</div>
```

CSS

```
css
.container {
  display: flex;
  gap: 20px; /* Mellomrom mellom seksjonene */
}

.section1, .section2, .section3 {
  flex: 1; /* Fordele plassen likt mellom seksjonene */
  padding: 20px;
  border: 1px solid #ccc;
}
```

- display: flex; på .container gjør at <section>-elementene plasseres horisontalt.
- gap gir mellomrom mellom seksjonene (alternativ til margin).
- flex: 1; fordeler plassen likt mellom seksjonene, slik at de får like mye bredde.



Koden min, frem til jeg spurte spørsmålet:

```
93
94     footer{
95         display: grid;
96         grid-row-start: 4;
97         grid-row-end: 5;
98         grid-column-start: 1;
99         grid-column-end: 4;
100         padding: 0;
101         margin: 0 auto;
102     }
103
104     #medieIkoner {
105         display: flex;
106         flex-direction: row;
107     }
108
109     #textBunn{
110         display: grid;
111         grid-row-start: 1;
112         grid-row-end: 5;
113     }
114
115     .sectionFooter{
116         display: flex;
117         flex-direction: row;
118         justify-content: space-between;
119     }
120
121
122
123
124
125
```

Koden min etter spørsmålet:

```
94     footer{
95         display: grid;
96         grid-row-start: 4;
97         grid-row-end: 5;
98         grid-column-start: 1;
99         grid-column-end: 5;
100         padding: 0;
101         margin: 0 auto;
102     }
103
104     #medieIkoner {
105         display: flex;
106         flex-direction: row;
107     }
108
109     #textBunn{
110         display: grid;
111         grid-row-start: 1;
112         grid-row-end: 5;
113     }
114
115     .sectionFooter{
116         display: flex;
117         gap: 20px;
118         justify-content: space-between;
119     }
120
121     .sectionFooter1 .sectionFooter2{
122         flex: 1;
123         padding: 20px;
124         gap: 20px;
125     }
126
```

Nytt spørsmål til ChatGPT:

Is fieldset the best way to make a border around an a element?

Using a `<fieldset>` element to create a border around an `<a>` (anchor) element is not the most appropriate or semantically correct method. `<fieldset>` is designed to group related elements within a form and typically includes a `<legend>` to describe the fieldset. It's generally used for grouping form controls, not for styling individual non-form elements like `<a>` tags.

Instead, you should use CSS to style the anchor element directly. There are several better approaches to adding a border around an `<a>` element, depending on the desired effect:

1. Using `border` in CSS

You can directly apply a `border` to an `<a>` tag:

html

Kopier kode

```
html<a href="https://example.com" class="bordered-link">Click me</a>
```

css

Kopier kode

```
.bordered-link {border: 2px solid #000; /* Add a solid black border */padding: 5px; /* Add padding to avoid the text touching the border */text-decoration: none; /* Optional: remove underline from link */}
```

2. Using `outline` for a focus-style border

If you want the border to appear when the link is focused (for accessibility), you can use the `outline` property. This is especially useful for accessibility because it helps with keyboard navigation.

css

Kopier kode

```
.bordered-link:focus {outline: 2px solid #000; /* Add a solid black outline when focused */}
```

3. Using `box-shadow` for a non-solid border effect

You can create a softer border effect by using `box-shadow`:

css

Kopier kode

```
.bordered-link {box-shadow: 0 0 0 2px #000; /* Creates a soft border effect */padding: 5px;text-decoration: none;}
```

4. Using `border-radius` for rounded corners

If you want rounded corners, you can use `border-radius`:

css

Kopier kode

```
.bordered-link {border: 2px solid #000;border-radius: 10px; /* Adds rounded corners */padding: 5px;text-decoration: none;}
```

↓

Summary:

Summary:

- Avoid using `<fieldset>` for non-form elements like anchor tags. It's semantically incorrect and can confuse screen readers.
- Use CSS (e.g., `border`, `box-shadow`, `outline`) to apply borders or other styles directly to the `<a>` element.
- Consider accessibility by ensuring links are focusable and providing visual cues for keyboard navigation.

🔊 📄 🌟 🗨️ ↺

Fant nylig utav når jeg endret litt på border, margin padding osv, for å få <a> elementet sentrert inne i en border, at kun den høyre delen av <a> elementet endret seg og da spurte jeg ChatGpt om hvorfor og fikk dette som svar:

when I change the width, why is it only the right side of a element that is moving and not on the right side?

Koden min før:

```
66
67
68 #sectionButtons li a{
69     margin: 1px;
70     width: 55px;
71     position: relative;
72     border: 1px solid black;
73     right: 42px;
74 }
75
76 #sectionButtons li{
77     display: flex;
78     justify-content: center;
79     width: 71px;
80     position: relative;
81     right: -63px;
82     height: auto;
83 }
84
85
86
```

Koden min etter:

```
67
68 #sectionButtons li a{
69     margin: 1px;
70     width: 55px;
71     position: relative;
72     border: 1px solid black;
73     right: 42px;
74     padding-left: 23px;
75 }
76
77 #sectionButtons li{
78     display: flex;
79     justify-content: center;
80     width: 71px;
81     position: relative;
82     right: -63px;
83     height: auto;
84 }
85
```

When adjusting the width of an `<a>` element, if only the right side appears to be moving, it could be due to the default behavior of inline or inline-block elements, or because of the way padding and margins are set. Here are some common causes and solutions:

1. Inline vs. Inline-Block Display

By default, `<a>` elements are inline elements, meaning they only take up as much width as needed for their content. This can cause only the right side to expand if you're changing `width`, since it doesn't affect the inline element's box.

To allow the width to apply uniformly, set the `<a>` element to `display: inline-block` or `display: block`:

```
css
a {
    display: inline-block; /* Allows width adjustments to apply uniformly */
    width: 200px; /* Set width as desired */
}
```

- `display: inline-block` will allow the element to remain inline with other elements but respect width and height settings.
- `display: block` will make the element take the full width of its container.

2. Padding and Margin on One Side

If you have `padding` or `margin` applied only to the right side, that could also cause a similar effect. For example, `padding-right: 10px;` will push only the right side out, giving the impression that only one side is moving. Ensure padding and margin are applied symmetrically:

```
css
a {
    padding: 10px; /* Adds padding uniformly around all sides */
    margin: 10px; /* Optional: Adds margin uniformly */
}
```

Or if you want specific padding:

```
css
a {
    padding: 10px 20px; /* 10px top/bottom and 20px Left/right */
}
```

3. Text Alignment and `text-align` Property

If the `<a>` element is within a container with `text-align: right` or similar styling, it may align only one side of the element. In such cases, you can either set `text-align: center` on the container or change the `<a>` to `display: block` to have it align uniformly.

4. Width Calculation and `box-sizing`

If you are using `width` and also setting padding or borders, make sure you're using `box-sizing: border-box;` to prevent extra padding from increasing the width asymmetrically:

```
css
a {
```

```
css
a {
    display: inline-block;
    width: 200px;
    padding: 10px;
    box-sizing: border-box; /* Ensures padding/border are included within width */
}
```

Summary

To control the width uniformly around an `<a>` element, use `display: inline-block` or `display: block`, ensure padding and margin are balanced, and consider setting `box-sizing: border-box`. This will ensure that width changes are applied equally to both sides.

Ikonene som jeg overførte ville ikke komme ut som hule, men heller fulle.

Å gjøre et ikon til hul eller “hollow”, har vi ikke gjort i timen og da kopierte jeg det ChatGPT skrev til meg:

how do I make the text hollow also that firefox can?

To create a hollow text effect that works in both Firefox and WebKit browsers like Chrome and Safari, you can use a combination of `text-shadow` and `-webkit-text-stroke` to achieve an outline effect without any fill.

Solution

1. Use `-webkit-text-stroke` for WebKit browsers (Chrome, Safari).
2. Use `text-shadow` to create a similar outline effect for Firefox.

Here's how you can set it up:

```
css
.hollow-text {
  font-size: 3em;          /* Adjust font size as needed */
  font-weight: bold;        /* Optional, makes the outline more pronounced */
  color: transparent;        /* Makes the text itself invisible */

  /* Webkit Browsers (Chrome, Safari) */
  -webkit-text-stroke-width: 2px; /* Thickness of the outline */
  -webkit-text-stroke-color: #000; /* Outline color */

  /* Fallback for Firefox and other browsers */
  text-shadow:
    2px 2px 0 #000, /* Shadow to the bottom-right */
    -2px 2px 0 #000, /* Shadow to the bottom-left */
    2px -2px 0 #000, /* Shadow to the top-right */
    -2px -2px 0 #000; /* Shadow to the top-left */
}
```

Explanation

- `color: transparent;`: This makes the main color of the text transparent so that only the outline is visible.
- `-webkit-text-stroke-width: 2px;` and `-webkit-text-stroke-color: #000;`: These properties outline the text in WebKit browsers (like Chrome and Safari). Adjust the width and color as needed.
- `text-shadow:`: Adds multiple shadows to simulate the outline effect in Firefox and other non-WebKit browsers. Each shadow is offset in a different direction (top-left, top-right, bottom-left, bottom-right) to create a complete outline around the text.

Complete HTML and CSS Example

```
html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hollow Text Effect</title>
  <style>
    .hollow-text {
      font-size: 3em;          /* Adjust font size as needed */
      font-weight: bold;        /* Optional, makes the outline more pronounced */
      color: transparent;        /* Makes the text itself invisible */

      /* Webkit Browsers (Chrome, Safari) */
      -webkit-text-stroke-width: 2px; /* Thickness of the outline */
      -webkit-text-stroke-color: #000; /* Outline color */

      /* Fallback for Firefox and other browsers */
      text-shadow:
        2px 2px 0 #000, /* Shadow to the bottom-right */
        -2px 2px 0 #000, /* Shadow to the bottom-left */
        2px -2px 0 #000, /* Shadow to the top-right */
        -2px -2px 0 #000; /* Shadow to the top-left */
    }
  </style>
</head>
<body>
  <h1 class="hollow-text">Hollow Text Effect</h1>
</body>
</html>
```

```
/* Webkit Browsers (Chrome, Safari) */
-webkit-text-stroke-width: 2px; /* Outline thickness */
-webkit-text-stroke-color: #000; /* Outline color */

/* Fallback for Firefox and other browsers */
text-shadow:
  2px 2px 0 #000, /* Bottom-right shadow */
  -2px 2px 0 #000, /* Bottom-left shadow */
  2px -2px 0 #000, /* Top-right shadow */
  -2px -2px 0 #000; /* Top-left shadow */
}
</style>
</head>
<body>

  <h1 class="hollow-text">Hollow Text Effect</h1>

</body>
</html>
```

Additional Tips

- **Adjusting Thickness:** You can change the `-webkit-text-stroke-width` and the `text-shadow` offset values (e.g., `2px 2px 0`) to make the outline thicker or thinner.
- **Outline Color:** Change `-webkit-text-stroke-color` and `text-shadow` color to adjust the outline color.

This approach should give you a reliable, cross-browser hollow text effect.

🔊 📄 🍷 🔄 🌐

Kopierte kode:

```
.fa-regular{
  /*Kopiert linjene under*/
  color: transparent;
  -webkit-text-stroke-width: 2px;
  -webkit-text-stroke-color: #000;
  /*Kopier linjene over*/
}
```

- Etter holdt på med media queries en stund, bestemte jeg for å øke overgangen fra 602px til 823px, da størrelsen til tlf ser bra ut, når jeg øker og minsker, men når jeg minsker desktop er det ikke samme måte, derfor tenker jeg at det beste alternativet er å øke.
- Etter å ha vurdert om å bruke randomisering av bilde innen anmeldelse seksjonen: (``) Kom jeg fram til at jeg plasserer denne koden her, og bruker heller et bilde innen filene "hund.jpg" slik at jeg viser god kompetanse om å linke til bilder lokalt.
- Jeg har gått mye frem og tilbake, ved å tolke eksamensteksten på forskjellige måter, ser derfor at commit'ene til github er litt rotete, men forståelig.
- Har også kommet over et problem, og det er på index.html, hvor det står ikke hvor avbrekket skal være mellom tlf og desktop, derfor har jeg satt tre avbrekk hvor det første på desktop vises slik at eksamensteksten forklarer at den skal se ut. Ved ned-eskalering av nettsiden vil så flex-wrap: wrap; bli forstyrret og dermed bryte formasjon, derfor det tredje avbrekket, da vil de plasseres under hverandre sentrert, også det siste avbrekket til overgang til tlf, vil da kategorikortene falle under hverandre og ligne på anmeldelsekortene, slik som teksten forklarer at jeg skal gjøre.
- Siden jeg også begynte på C-kravet, har jeg lagt til ikon for > istedenfor et symbol og har ikke skiftet tilbake, da dette viser at jeg har kompetanse til å legge inn ikoner i en brødsmulevei.