

Hvordan posisjonere en DIV

Hva er en DIV? En DIV er en “beholder” som inneholder innhold på en nettside. En div er noe som blir definert i CSS med enten en ID eller en Class. En ID/Class gir DIVen attributter og disse kan definere posisjonen til DIVen. De tre måtene man kan posisjonere en DIV på er ved hjelp av Transform, Grids og FlexBox.

Transform

For illustrasjoner se [modell 1](#)

Når vi bruker transform så setter man DIVen sin **position:absolute**; og definerer hvor på skjermen den skal være i forhold til kantene i nettleseren.

Fordeler med transform: med å bruke transform er at du slipper å bruke en “container” for å definere plasseringen til DIVen.

Ulemper med transform: Jeg vil ikke anbefale denne måten å posisjonere på fordi når man bruker transform og setter posisjonen til **left:50%**; og **top:50%**; så setter den det øvre venstre hjørnet av DIVen i senter av skjermen. Hvis man ønsker å ha midten av DIVen i midten av skjermen må man bruke **transform:translate(-50%, -50%)**; for å sette den tilbake 50% av sin egen høyde og bredde for å få den helt sentrert. Når man bruker transform så må man definere et nytt element hver gang noe skal legges til. I motsetning til som har ein container som man kan legge elementer til i og er sier man baer hvilken rad og kolonne den skal være i. Det største problemet til transform er at den ikke er like tilpasningsdyktig når det er forskjellige skjerm og nettleser størrelser.

Grids

For illustrasjoner se [modell 2](#)

Når vi bruker grids så setter vi opp et rutenett som vi kan sette DIVs inni. Måten man gjør dette på er at man definerer hvor mange rader og kolonner som det skal være i rutenettet. En ulempe med grids er at man må legge til et ekstra element i denne “grid-containeren”. På tross av at man må lage en container så blir det lettere å utforme siden som man ønsker. Med grids så slipper du å tenke på å tilpasse

nettsiden til forskjellige plattformer fordi grids fikser det for deg. Et annet stort pluss med grids er at man lett kan legge til mer innhold på nettsiden og justere på utseende i “containeren” uten store forandringer i griden. Det største problemet med grids er at det er litt vel komplisert å sette seg inn i når man starter med HTML. Det som gjør det vanskelig å sette seg inn i er at en del av grid oppsettet er kolonner men det er ikke veldig intuitivt å kontrollere dem.

Hvis du fremdeles ønsker å jobbe med grids så kan de lese guiden til Kristoffer(legger inn en link her når jeg har muligheten til det.)

FlexBox

For illustrasjoner se [modell 3](#)

Når vi bruker flex så setter vi opp rader og kolonner. Når man bruker flex så har man en hoved container som enten kan gå vertikalt eller horisontalt. Inne i hoved containeren så kan man sette mindre containere som går motsatt vei av hoved containeren sin retning. I eksempelet så har jeg en sentrert kolonne som som går nedover også har jeg brukt **Justify-content:center**; for å ha den midt i kolonnen. Noen fordeler med flex er at det er veldig lett å posisjonere flere DIVER i forhold til hverandre og det er veldig responsivt. De problemene som er litt gjennomgående er at det blir veldig mange containers og fra tid til annen så forskyver elementene på hverandre.

Oppsummering

Jeg vil anbefale å bruke både grids og flexbox overfor transform på grunn av deres fleksibilitet og tilpasningsdyktighet. Det beste er å bruke en kombinasjon av grids og flexbox for å ha tilgjengelig en del av **Justify** og **Align** fra flexbox for å kunne justere innholdet i en DIV og et godt overordnet system med grids for å velge hvor DIVene skal være.

Modell

Yo ikke bruk transform

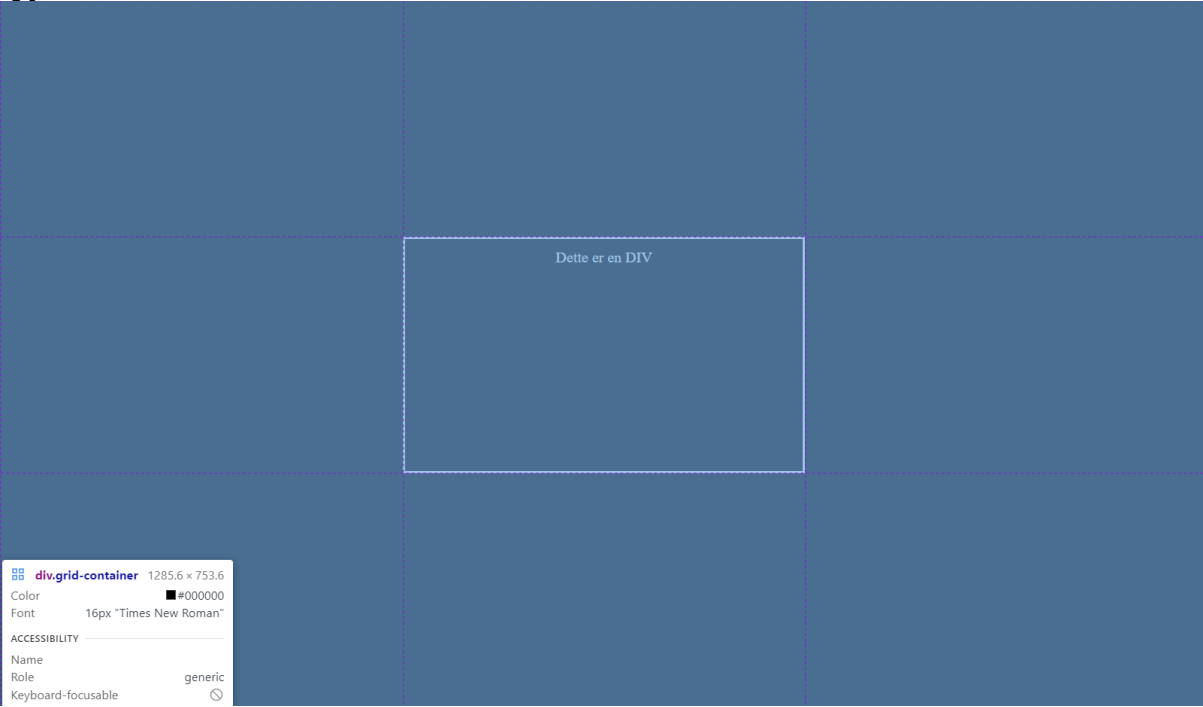
```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="UTF-8">
  <title>DIV</title>
</head>
<body>
  <div class="divTransform">Yo ikke bruk transform</div>
</body>
</html>
```

```
.divTransform{
  color: white;
  border: 3px solid white;
  position: absolute;
  width:400px;
  height: 100px;
  top:50%;
  left:50%;
  transform:translate(-50%, -50%)
}
```

```
}
```

Modell

2f



```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="UTF-8">
  <title>DIV</title>
</head>
<body>
  <div class="grid-container">
    <div class="divGrid">Dette er en DIV</div>
  </div>
</body>
</html>

```

```

.grid-container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 1fr;
  height: 100vh;
}

.divGrid{
  color: ■white;
  grid-column: 2/3;
  grid-row: 2/3;
  padding-top: 10px;
  justify-content: center ;
  border: 3px solid ■white;
  height: 1fr;
  width: 1fr;
}

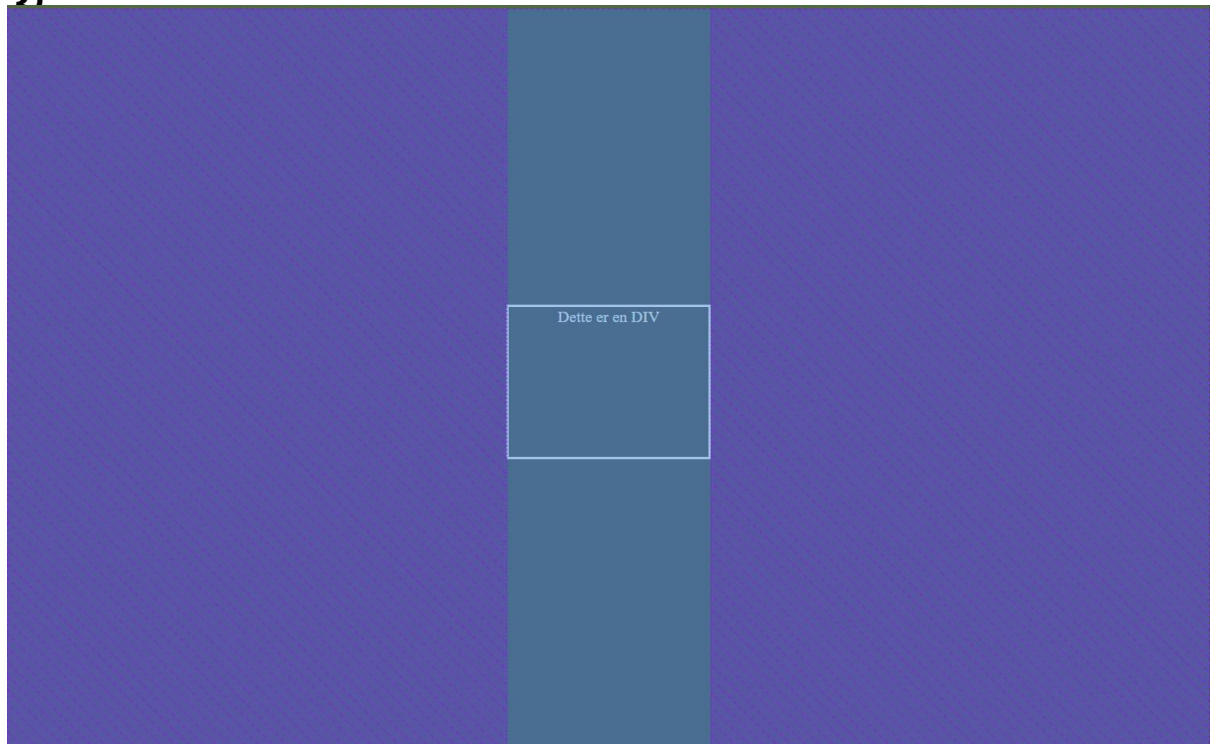
```

```

}
```

Modell

21



```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="UTF-8">
  <title>DIV</title>
</head>
<body>
  <div class="flex-container">
    <div class="divFlex">Dette er en DIV</div>
  </div>
</body>
</html>

```

```

.flex-container{
  display: flex;
  height: 100vh;
  padding-top: 5px;
  margin: 0;
  align-items: center;
  justify-content: center;
}

```

```

.divFlex{
  color: white;
  border: white solid 3px;
  height: 150px;
  width: 200px;
}

```

```

}

```