

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

# Cascading Style Sheet

Basic Page Structure



# Viewport

## Meta viewport element

- Browseren er automatisk ikke sat til at være responsiv, der skal derfor indsættes et meta viewport i HTML koden.

## Meta Viewport attributes

- Meta viewport har 2 attributes, som skal tilføjes for at gøre browseren responsiv.
- Attribute 1: **width=device-width** - *Fortæller browseren hvordan den skal skalere (px).*
- Attribute 2: **initial-scale=1. (1 px = to 1 device independent px)** - *Sørger for at hjemmesiden ikke zoomes, når den loades.*

## Bad practice

- User-scalable
- minimum scale
- maximum scale

Brug af disse properties stopper brugeren for at zoome ind og ud på skærmen, og det er u hensigtsmæssigt for brugeren, derudover er det også en nødvendighed for accessibility, at brugeren har mulighed for at zoome ind eller ud.

**Bonus:** Der kan være i web-app spil, hvor man ikke vil have, at brugeren kan zoome ind og dermed bruger disse properties.

# Required CSS

## Box model

Alle HTML elementer anses automatisk som at være i en boks (box-model).

Box-model indeholder altid: content (indholdet), padding, border og margin.

**Box-sizing: content-box:** Indlæser **automatisk** kun indholdet, og vil ikke tage højde for padding og border.

**Box-sizing: border-box:** Definerer størrelsen på boksen, hvis man bl.a. indsætter width og height properties, og vil derfor tage højde for indholdet, padding og border.

```
*, *:before, *:after {
```

```
  box-sizing: inherit;
```

```
}
```

= Gør at kasserne/stylingen bliver kopieret til alle tekstelementer på siden i samme størrelse.

## Billeder i RWD

**Max-width: 100%** - Sikrer at et billede skalerer sig så det passer med størrelsen af viewporten

```
<!DOCTYPE html>
<html>
<head>
<style>
html {
  box-sizing: border-box;
}

*, *:before, *:after {
  box-sizing: inherit;
  width: 300px;
  height: 100px;
  border: 10px solid blue;
}
</style>
</head>
<body>
```

Defines how the width and height of an element are calculated; should they include padding and borders, or not.

**box-sizing: content-box (default):**

Width and height only apply to the content of the element:



# Display Property

## Display

- **inline;** Bruges, hvis man ønsker, at ændre en del af en linje i en tekstdel. Inline vil ikke 'break' flowet af den anden boks. Man kan tilføje padding og margin, men det vil stadig respondere til det padding og margin der er udenom den anden boks. Vil også ignorere width og height properties.
- **block;** Et element der tager hele 'delen' af et element. F.eks. H1, div, section eller nav. Block elementer tager automatisk den fulde width og height, der er tilgængelig på siden medmindre man definerer dette. Et block element vil altid starte på en ny linje 'by default'.
- **inline-block;** En kombination af inline og block. Inline-block vil fremvises i linje med et andet element, men inline-block kan få dets egen definition af width og height. Det vil dog stadig stå på samme linje, som den tekst det er omringet af.



# Positioning

## Position:

- **static**; Elementerne er altid automatisk positioneret **static**.
- **relative**; Relative flytter elementerne til en ny lokation ud fra sin oprindelige lokation - dvs left: 40 px flytter elementet 40 px til højre
- **absolute**; Flytter elementet til den lokation der specificeres. Men denne efterlader ikke et tomt felt hvor dens oprindelige placering var - som de andre position properties gør. Den sætter sig i stedet for ovenpå de andre elementer på siden. Når man flytter elementet som absolute henviser den desuden til bodyen - dvs fra øverste venstre hjørne af skærmen, og ikke fx den div den står i.
- **fixed**; Bruges, hvis man ønsker, at et element fastholdes på siden. Fixed sidder fast hele tiden.
- **sticky**; Bruges, hvis man ønsker, at et element fastholdes på siden. Sticky differentiere sig fra fixed ved at den "fanger" elementet når man scroller ned til det på siden - og her sætter sig fast som fixed. Sticky kan f.eks. bruges til at fastholde en navigationsbar, når brugeren scroller ned på siden.

\*Et element der er positioneret, vil altid være foran et element som ikke er positioneret. Man kan ændre hvilket element, der skal være foran eller bagved ved at bruge **z-index: 1**; (foran) eller **z-index: -1**; (bagved). Man kan bruge z-index, hvis man har gjort sit element til display; relative, absolute, fixed, sticky, da det **ikke** kan anvendes ved value static. For at sætte et ikke-positioneret-element foran/bagved giver man det blot en relative positionering uden px ændringer.

# Floats

- En tidligere metode, floats. Før flex og grid
- Float, bruges til at placere et element til venstre eller højre side, som gør at andre elementer kan 'wrappes' rundt om det.
- Det blev oprindeligt lavet til at få billeder til at have tekst ved siden af billedet (boksen).
- **clear:** bruges til at starte et nyt afsnit efter en billedtekst f.eks., for at afslutte float.

*\*Floats bruges ikke længere hvis man laver en ny hjemmeside, men kan ses hvis man kigger på eksisterende hjemmesider.*



```
7  <!-- SCREENSHOT: A LAYOUT WITH A PHOTO, A COLUMN OF TEXT, AND A COLUMN OF TEXT -->
8  <style>
9  .photo {
10     float: left;
11     margin-right: 20px;
12     margin-bottom: 20px;
13 }
14 .p3 {
15     clear: left;
16 }
17
18 </style>
```



# Units

## Absolute unit

- Pixels - bruges ofte til websites. En boks ændrer ikke størrelse hvis man ændrer på viewporten.

## Relative units - **Responsive**

- **Procent %** - Baseret på 'height' og 'width' af et element. Brugen af procent, vil altid respondere til hvor stor skærmstørrelsen er.
- **vw** - Står for 'viewport width' - den beskriver 1/100th -> dvs. en vw på 50 = halvdelen af skærmens bredde.
- **vh** - Står for 'viewport height' 1/100th af højden
- **vmin** - Relativ til 1/100th af viewports mindre dimension. Dvs. er bredden af skærmen den mindste, så bliver elementet med fx 50vmin til 50% af skærmens bredde - selvom længden er meget længere. Den tager ud fra den mindste dimension.
- **vmax** - Relativ til 1/100th af viewports største dimension - modsat vmin.
- **fr** - fractional unit er en brøkdel af viewport størrelsen. Bruges kun i grid layouts.