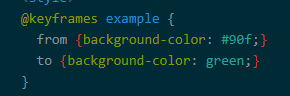
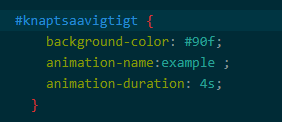
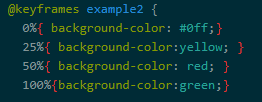
**Animationer og transitions i CSS**

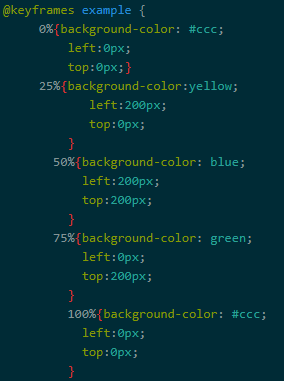
1. **Sæt jer ind i emnet med brug af en bred vifte af kilder.**  
   CSS3 animationer gør det muligt at animere overgange fra en CSS style konfiguration til en anden. Animationer består af to komponenter: En styling der beskriver animationen og et sæt af keyframes, der angiver start og slut stadiet for Animationen. Derudover kan der også være mellemliggende waypoints undervejs. (<https://www.nemprogrammering.dk/Tutorials/CSS3/8_animationer/animationer_CSS3.php>)   
   (<https://www.w3schools.com/css/css3_animations.asp> )  
   (<https://www.nemprogrammering.dk/Tutorials/CSS3/7_transitions/transitions_CSS3.php>)
2. **Udarbejd en række eksempler, der på pædagogisk vis viser:**  
   **- Hvordan man arbejder med det i praksis  
   - Bredden af (relevante) muligheder**
3. **Skriv et kapitel til en webbaseret lærebog:**  
   **- Skal indeholde en introduktion til emnet + skærmbilleder   
   - Forklarende tekst  
     
   Animationer:**   
   CSS animationer gør det muligt at animere overgangene fra en CSS style til en anden. Dette kan fx være man har en boks på hjemmesiden, som man gerne vil ændre farven eller størrelsen på ved hjælp af en animation.  En animation er en samling af keyframes og man kan i hvert keyframe specificere CSS egenskaber. For at lave en animation skal man i eksempelvis ”Brackets” under CSS-filen bruge nøgleordet @keyframes. Her er et eksempel på hvordan kodningen bag en animation kan se ud:   
     
   Her skifter man baggrundsfarven til en anden farve via en animation.

Det er så den her ens “@keyframes” hænger sammen med, netop fordi man har givet den her “@keyframes” et navn som i dette tilfælde er “example”. Denne her form for id eller navn man har givet keyframes kan man se komme igen nede i tagget “#knaptsaavigtigt” dette gør at ens @keyframes kan finde frem til den rette del af koden som den skal tilføje sig selv til. I keyframes kan man enten bruge “0%” eller “100%” for at definere en start og en slutpunkt, hvor 0% er startpunktet og 100% er slutpunktet. Man kan også bruge “from” og “to” for at definere en start og et slutpunkt.   
  
Animation-duration beskriver hvor lang tid selve animationen skal være. I dette tilfælde vil det være 4 sekunder.

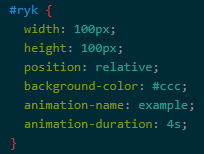
Eksempel to, har vi brugt % til at definere de forskellige stadier som vores animation gennemgår. Dette hjælper os til at kunne give vores objekt flere animationer på en duration. I vores tilfælde har vi givet objektet fire forskellige farver i løbet af durationen.



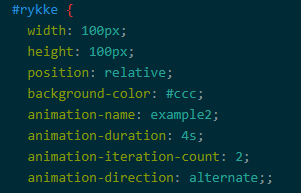
I den her keyframe kan vi se at farven er kodet #ccc, som i dette tilfælde er grå som startpunktet 0%. I den første ser vi også nogle koordinater, de koordinater er start punktet for figuren, da det befinder sig i startpunktet. I den næste har vi 25% som har farven gul, ud over kan man se at koordinaterne har ændret sig, altså at figuren har rykket sig, hvor i vores tilfælde er det 200px(pixel) til venstre, men befinder sig stadig 0px i forhold til toppen. Til den tredje del af figuren, har man så valgt at give den farven blå, hvor den har koordinaterne 200px til venstre og 200px i forhold til toppen. Til den fjerde har man valgt at bruge farven grøn som baggrundsfarve som del af animationen der tilhøre figuren. Denne gang har figuren så rykket sig til at have 0px til venstre, men stadig har 200px i forhold til toppen. Til den sidste bliver baggrundsfarven den samme som den var til start, og det er fordi det er her slutpunktet for animationen er. Det er også derfor at både top og til venstre har samme lokation i forhold til hinanden.



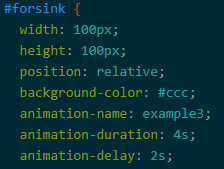
Vi har den første af rykke funktionerne, i dette tilfælde har vi angivet en af sections med id’et “ryk” for at kunne bruge det som tag til den første objekt der kan flyttes rundt med.  De to første ting vi har er dem der hedder width og height. og det er dem der bestemmer højden og bredden af vores figur, som i dette tilfælde er 100px ved begge tags. Den næste er baggrundsfarven som er grundfarven til figuren. Det betyder at den vil have den som standard, med andre ord så vil den have den fra start og når animationen slutter igen, hvis vi altså har angivet andre farver i keyframes. Den næste nye tag vi har en “position” som i dette tilfælde er relativ til det punkt figuren startpunkt er.



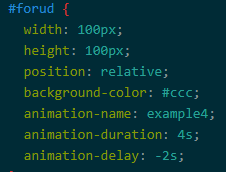
De næste to nye som man kan se er blevet brugt til denne animation er “animation-iteration-count” som betyder hvor mange gange den her animation skal køre. Og den anden er “animation-direction” som kan lave om  hvilken vej den tager af den tur vi har givet den. I dette tilfælde har den fået “alternate” som betyder at den køre begge vej, først den oprindelige del af vejen, og derefter baglæns. Man kan også indsætte værdien “reverse” i “animation-direction”, hvor denne værdi gør sådan at figurens animation foregår baglæns. Man kan vælge at bruge værdien “alternate-reverse” som betyder at den først starter med at køre baglæns i animationen og derefter forlæns(altså den rigtige vej til sidst).



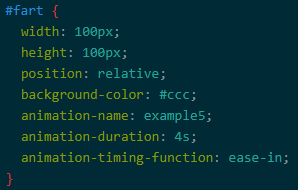
Til eksempel tre her, så har vi en ny tag der hedder “animation-delay” som siger hvor lang tid skal der går fra at animationen bliver startet til at den reagere, hvor i vores tilfælde er det 2 sekunder.



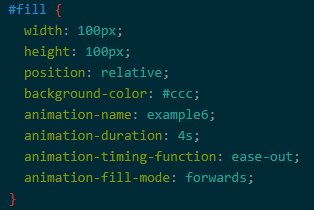
Det næste eksempel her, altså eksempel 4, har vi det samme tag “animation-delay” men denne gang har den fået værdien -2s, altså minus 2 sekunder, hvilket der gør at figuren starter 2 sekunder inde i selve animationen, hvilket også betyder at i stedet for at animationen vare i 4 sekunder, så vare det kun i 2 sekunder.

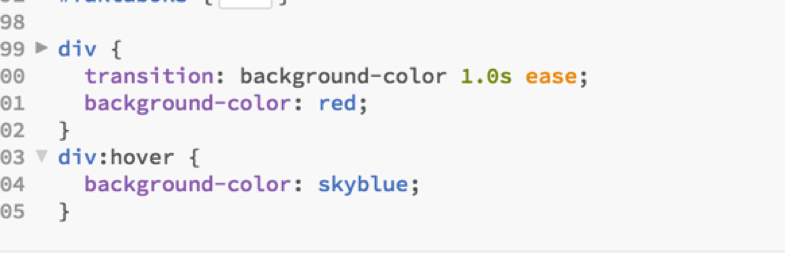
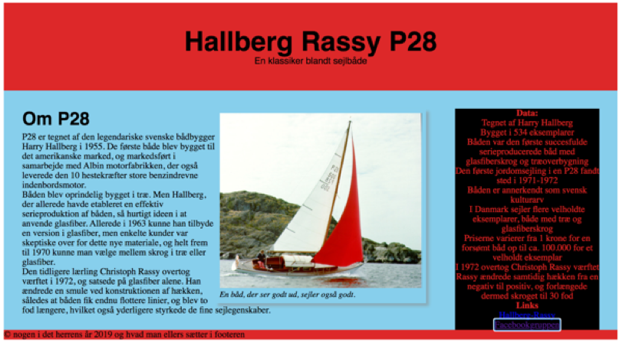
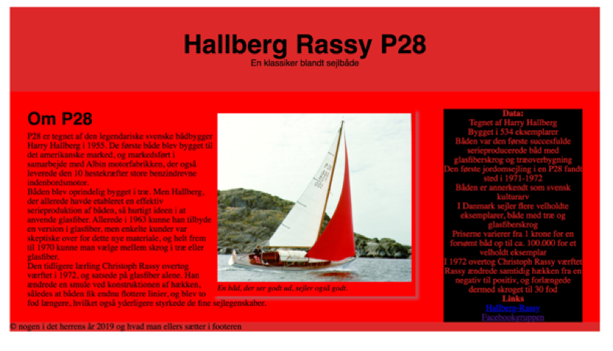


Den nye funktion som hedder “animation-timing-function” er tag der gør at man kan bestemme farten på figuren, samt hvorhenne i animationen den skal have den her specifikke animation. I dette tilfælde har den fået værdien “ease-in” som betyder at den har en langsom start. Man kan vælge at bruge værdien “ease-out” som betyder at den har en normal fart til at starte med, men vil have en langsom fart til sidst inden animationen stopper. Værdien “linear” gør sådan at figuren har den samme fart hele vejen igen animationen. “ease” er værdien der gør at figuren har en langsom start, hvor den derefter får en højere fart, og til sidst slutter med en langsom fart(dette er hvad den har som standard). Den næste værdi er “ease-in-out” som betyder at figuren har en langsom start og en langsom slutning i animationen.



Det sidste animation tag vi arbejdede med var “animation-fill-mode” som har den funktion at den kan specificere en “style” for det valgte element før elementets animation går i gang, men også efter at elementets animation er ovre. “animation-fill-mode” har fire værdier, som er “none” “forwards” “backwards” og til sidst “both”.  Værdien “none” betyder at den har den normale værdi for elementet, altså er det den standard den plejer at køre med. Værdien “forwards” har den betydning at elementet vil beholde de “style values”  som var sat til det sidste del af keyframe(dette er så alt afhængig af at “animation-direction” og “animation-iteration-count”).  Værdien “backwards” har den funktion at den kan få de værdier for elementets startpunkt i animationen for den valgt keyframe(er dog afhængig af “animation-direction”) hvor den samtidig beholder den værdi i forhold til “animation-delay”. I vores tilfælde har vi valgt give vores “animation-fill-mode” værdien “forwards”.



**Transitions:**CSS overgange/transitions gør det muligt at bestemme over overgangene på hjemmesiden. Dette kan eksempelvis være skygger, som kommer frem, hvis man holder musen over en titel. Det kan også være en boks på siden, som skifter farve. Det er præsentations effekter, som gør at værdierne på hjemmesiden bliver ændret. Disse transitions giver programmøren mulighed for at bestemme en overgang, så den ændrer sig i det tempo og på den måde, som programmøren af hjemmesiden ønsker. Boksen der blev nævnt tidligere, kan altså skifte farve i præcis det tempo, som programmøren vælger. Overgangseffekterne kan benyttes på mange af CSS egenskaberne fx baggrundsfarve, bredde, højde, opacitet osv. Disse transitions gør det muligt, at give hjemmesiden præcis den udstråling, som programmøren finder passende. Det er med til at forme brugeroplevelsen af hjemmesiden.   
  
Forskellen på animation og transitions ligger i hvordan man skriver det i CSS. Kort fortalt er animationerne et lag oven på transitions, som giver endnu mere kontrol og flere muligheder. Animationerne er altså en udvidet form for transitions.    
  
Eksempler på transitions:   
  
EKSEMPEL 1:  
Hjemmesiden inden man holder musen hen over siden, og den blå er efter man holder musen henover siden.  
  
  
EKSEMPEL 2:  
