## Lektion 3 – HTML, CSS och JavaScript

I den här lektionen behandlas HTML, CSS och JavaScript (JS). Det förutsätts att lektion 1 och 2 är gjord, eller att du har tillräckliga kunskaper i grundläggande HTML och CSS för att klara den här lektionen.

I denna lektion kommer JS att användas för att förändra innehållet i en webbsida (dvs HTML), och för att förändra utseendet (dvs CSS).

HTML - Börja så här

- 1. Skapa ett nytt html-dokument.
- 2. Skriv Lektion 2 som titel.
- 3. Sedan ska själva strukturen, fyrkanterna, för webbsidan göras. Dessa fyrkanter skapas med hjälp av <div>. En div är ett generiskt element som kan innehålla andra element, och som är lätt att placera på webbsidan. Läs mer om div här: http://www.w3schools.com/html/html\_layout.asp http://www.w3schools.com/tags/tag\_div.asp
  Skapa fyra divar med följande id: left mid\_right\_och textBox\_Dessa ska tillhöra

Skapa fyra divar med följande id: left, mid, right, och textBox. Dessa ska tillhöra en klass som kan kallas content. Koden bör nu se ut så här ungefär:

I textBox ska ett stycke läggas till med siffran 0 i. Ge stycket id't clickCounter, vilket kommer att utnyttjas i JS senare.
 <div id="textbox">0</div>

CSS - Fortsätt så här

- 1. En div som inte har något innehåll syns inte på webbsidan. För att kunna se vad som gjorts hittills i lektionen, behövs en CSS-fil skapas. I CSS-filen ska body få #6a6a6a som bakgrundsfärg. #left och #right ska få #a6a6a6 som bakgrundsfärg och mid ska få #ff7d00 som bakgrundsfärg. #textBox behöver ingen bakgrundsfärg utan bakgrundsfärgen från body får gälla.
- 2. Klassen content ska få bredd respektive höjd till 20vw. Det betyder att varje element som tillhör klassen content ska vara 20% av viewport width i höjd och i bredd. Viewport width innebär bredden på browserfönstret. Detta är ett bra sätt att få innehållet på webbsidan att bli mer responsiv eftersom storleken på elementen anpassas efter storleken på fönstret. Läs mer här:

```
https://css-tricks.com/viewport-sized-typography/
         Så här bör CSS-koden se ut ungefär:
body {
      background-color: #6a6a6a;
}
.content {
    width: 20vw;
    height: 20vw;
#textBox {
}
#left {
      background-color: #a6a6a6;
}
#mid {
      background-color: #ff7d00;
}
#right {
      background-color: #a6a6a6;
```

- }
- 3. Om CSS-filen sparas och webbsidan laddas kommer en mörkgrå bakgrund användas. Till vänster i fönstret finns en ljusare grå ruta, med en orange ruta under, under den en ljusgrå ruta, och längst ned siffran 0.
- 4. Detta är helt ok, men de tre färgade rutorna ska ligga horisontellt bredvid varandra. Detta görs genom att lägga till float: left; i CSS-filen under .content. Float innebär att elementen flyter och hamnar bredvid varandra justerade åt vänster. För att inte siffran 0 också ska flyta och justera sig åt vänster, men till höger om fyrkanterna måste clear: left användas i #textBox. Läs mer om float här: http://www.w3schools.com/css/css\_float.asp
- 5. Nu är det dags att snygga till fyrkanterna. Först ska de få marginal mot omkringliggande element med margin: 2vw;.
- 6. Nästa steg är att göra fyrkanterna mindre kantiga genom att ändra border-radius till 2vw. Detta rundar av hörnen. Läs mer här: http://www.w3schools.com/cssref/css3\_pr\_border-radius.asp
- 7. Sedan ska fyrkanterna få en vit kant/ram runt om. Det görs med border, och ramen ska vara solid 2px tjock och vit. Det skrivs så här: border: solid 2px #fff;. Läs mer om border här:

http://www.w3schools.com/css/css\_border.asp

8. Därefter ska rutorna få en skuggning mot bakgrunden. Detta görs med boxshadow. Box-shadow har fem saker att specificera: horisontell skugga, vertikal skugga, suddighet i skuggan, spridning av skuggan, samt färg på skuggan. Färgen sätts som RGB-värde med komma mellan varje färgkanal samt alphakanalen (genomskinligheten). Sätt skuggan så här: box-shadow: 0 6px 12px 3px rgba(0, 0, 0, 0.4);. Läs mer om skugga här:

http://www.w3schools.com/cssref/css3 pr box-shadow.asp

9. Spara CSS-filen och ladda om webbsidan. Nu syns fyrkanterna som de ska, även om den undre inte är placerad på rätt ställe. Detta fixas genom att ändra marginalinställningarna för #textBox. Testa margin: 2vw 26vw; vilket ser ok ut, även om det inte blir helt perfekt.

10. Nu är det dags att se över siffran 0. I p i CSS-filen ska teckensnittet ändras med font-family till 'Open Sans Condensed', sans-serif;. Sätt storleken till 8vw, fontvikten (dvs smal via normal till fet) till 300, färgen till orange, och centrera texten. Så här:

```
p {
```

```
font-family: 'Open Sans Condensed', sans-serif;
font-size: 8vw;
font-weight: 300;
color: #ff7d00;
text-align: center;
```

}

Läs mer här: http://www.w3schools.com/cssref/pr\_font\_font-family.asp http://www.w3schools.com/cssref/pr\_font\_font-size.asp http://www.w3schools.com/cssref/pr\_font\_weight.asp http://www.w3schools.com/cssref/pr\_text\_text-align.asp

- 11. Nu ser 0:an bättre ut, även om den sitter lite för långt ned i rutan. Detta ordnas genom att förändra radhöjden för text i diven texten sitter i, dvs textBox i detta fall. Ange: line-height: 5vw;. http://www.w3schools.com/cssref/pr\_dim\_line-height.asp
- 12. Det fyra fyrkanterna ska vara klickbara, och det ska visas genom att musmarkören får ett nytt utseende när den är över fyrkanterna. Detta åstadkoms med cursor: pointer;. Läs mer om cursor här: http://www.w3schools.com/cssref/pr class cursor.asp
- JS Äntligen är det dags
  - 1. Det går att skriva JS direkt i HTML-dokumentet, men det ska vi inte göra. Det är bättre att samla en eller flera funktioner i en egen fil. Denna fil länkas till från headern i HTML-dokumentet precis som till CSS-filen. På så sätt kan flera HTMLdokument använda samma uppsättning JS i stället för att samma JS ska skrivas i varje HTML-dokument. JS-filen kan i sin tur innehålla flera funktioner. I stället för link används script och vilken källa (src) som avses. Script-taggen måste också avslutas till skillnad mot link. Så här: <script src="lektion2.js"></script>
  - 2. Kommentarer i JS skrivs för en rad med // följt av kommentaren, eller för flera rader /\* följt av ett antal rader med kommentarer, och avslutas med \*/.
  - 3. Skapa nu en ny textfil. Döp den till lektion2.js.
  - 4. En funktion i JS skrivs som function namnetPåFunktionen(eventuella variabler för att ta emot data) { och avslutas sedan med }. Till exempel så här:

```
function changeColor(id){
    alert(id);
```

}

5. Innan det skrivs något i JS behövs HTML-filen ändras något. Fyrkanterna left, mid, och right ska ropa på en funktion som heter changeColor. Det görs med onclick, så här: onclick="changeColor(this.id). Det innebär att när något klickas på anropas funktionen changeColor och id't på elementet som klickades på skickas med. Läs mer här:

http://www.w3schools.com/jsref/event\_onclick.asp

- 6. Nu är det dags att skriva JS. Funktionen ska heta changeColor och den ska ta emot id't och spara det i en variabel som kallas id. Sedan ska en alert visas som i sin tur visar id't på elementet som klickades på. En alert visar en ruta, ett nytt litet browserfönter, med textinformation till användaren. Denna alertruta har en Ok-knapp. En alert är också ett ganska smidigt sätt att använda för felsökning, då till exempel olika alerts kan visas beroende på var man är i en if-sats eller för att visa innehållet i en variabel. Läs mer om alerts här: http://www.w3schools.com/js/js popup.asp
- 7. Spara JS-filen, och ladda om webbsidan och testa att det fungerar. Om allt fungerar som det ska är det dags att ta bort alerten från funktionen, och börja koda JS på allvar. Det första som ska göras är att låta JS läsa in värdet som finns i siffran som visas på webbsidan, dvs i id="clickCounter". Detta värde ska sparas i en variabel. En variabel i JavaScript deklareras som var variabelnsNamn = värde; (till exempel: var x = 5;). Tänk på att alltid använda bra variabelnamn som visar vad variabeln faktiskt används till, detta kommer att underlätta all kodning och felsökning. Läs mer om variabler i JS här:

http://www.w3schools.com/js/js\_variables.asp

8. Sedan ska värdet i clickCounter läsas på webbsidan. Det görs med följande kod: document.getElementById("clickCounter").textContent. Det innebär att textinformationen (textContent) i clickCounter (getElementById("clickCounter")) hämtas från webbsidan (document). Läs mer här:

http://www.w3schools.com/jsref/prop\_node\_textcontent.asp

9. Då kommer siffran att läsas ut till en sträng. En sträng går inte att räkna matematik med varför strängen behöver göras om till ett heltal (integer). Genom att använda parseInt görs siffran i strängen om till ett heltal. Läs mer här: http://www.w3schools.com/jsref/jsref\_parseint.asp

https://developer.mozilla.org/en-US/docs/Web/JavaScript/

Reference/Global\_Objects/parseInt

Läs mer om strängar här: http://www.w3schools.com/js/js\_strings.asp Detta ger en kodrad liknande denna:

```
var numberOfClicks =
```

parseInt(document.getElementById("clickCounter").textContent);

10. Nu läses siffran från webbsidan, men varje klick som görs på rutorna ska göra att denna siffra uppdateras. Det görs genom att använda i stort sett samma kod som tidigare, men denna gång ska clickCounter få värdet av numberOfClicks + 1. Så här:

document.getElementById("clickCounter").textContent = numberOfClicks+1;

11. Spara JS-filen och ladda om webbsidan. Nu ska siffran uppdateras varje gång någon av de tre fyrkanterna klickas på. Det enda sättet att åter nollställa siffran är dock att ladda om sidan. Det är tråkigt. Därför ska rutan med siffran i göras klickbar och den ska kalla på (med hjälp av onclick) en ny funktion som kallas resetPage. Ändra i HTML-koden så att textBox har en onclick. Skriv sedan en ny funktion i JS-filen som sätter clickCounter = 0. 12. Nu ska rutorna byta färg och den som klickas på ska bli orange medan de övriga ljusgrå. Detta kan göras på olika sätt, men i detta fall ska en if-sats i funktionen changeColor användas. En if-sats kollar om ett specifikt villkor uppfyllts och om så är fallet ska vissa saker ske eller ändras. En if-sats skrivs som if (villkor) { vad som ska ske }.

I detta fall: Om left klickas på ska den bli orange och då ska mid samt right bli Ijusgrå. Om mid klickas på ska den bli orange och de andra Ijusgrå. Om right klickas på ska den bli orange medan de andra blir Ijusgrå. Detta skulle kunna skrivas som tre if-satser, men det är bättre att skriva en if-sats som innehåller else if (annars om). Villkoren för else if kollas om villkoren i den första if inte uppfyllts. Sist kan man ha en else (annars). Uppfylls inte villkoren i tidigare kontroller så gäller alltid else. Så här kan vi skriva if-satsen:

```
if (id == "left") {
```

```
} else if (id == "mid") {
```

```
} else {
```

}

Detta fungerar eftersom det bara är left, mid eller right som anropar funktionen. Och är det left som klickades ska left bli orange, eller om det är mid så ska mid bli orange, annars är det right som klickades och då ska right bli orange. Läs mer om if-satser här:

http://www.w3schools.com/js/js\_if\_else.asp

För att kunna skriva if-satser korrekt måste man känna till hur operatorerna för jämförelser skrivs. Lika med, som ska användas i detta fall, skrivs med två ==. Läs mer om operatorer här:

http://www.w3schools.com/js/js\_comparisons.asp

13. Nu ska färgen på rutorna ändras. Det görs på samma sätt som för siffran som räknas upp, men i stället för textContent ska backgroundColor ändras. Det görs först genom att påpeka att det är CSS som ska ändras med style följt av backgroundColor. Det id som klickas på ska få "#ff7d00" medan övriga id ska få "#a6a6a6". Varje del av if-satsen behöver alltså få tre rader som sätter färgen för varje fyrkant.

document.getElementById("left").style.backgroundColor = "#ff7d00";

14. Spara JS-filen och ladda om sidan och kolla att allt fungerar. Ändra i funktionen resetPage så att fyrkanternas bakgrundsfärger återställs med orange i mitten och ljusgrått till vänster och till höger. Sedan är all kod JS färdigskriven.

CSS - En sista finess

1. Avslutningsvis ska en sista finess göras i CSS-filen. Det som ska ändras är att ändringen av bakgrundsfärg i de tre fyrkanterna ska göras med en snabb fade (transition). Övertoningen (faden) mellan orange och ljusgrått, och vice versa, ska vara snabb, 0.2 sekunder. Problemet med transition är att standarden inte är helt bestämd, utan varje webbrowser tolkar koden lite olika. Därför blir vi tvungna att skriva flera rader kod för att vara säkra på att övertoningen ska fungera bra. Övertoningseffekten ska ligga i klassen .content för att påverka alla fyrkanterna. -webkit-transition: background-color 0.2s ease-in-out;
-moz-transition: background-color 0.2s ease-in-out;
-chtml-transition: background-color 0.2s ease-in-out;
transition: background-color 0.2s ease-in-out;
Läs mer här:

http://www.w3schools.com/css/css3\_transitions.asp
Och här om browserkompatibilitet:
http://caniuse.com/#feat=css-transitions

2. När all kodning är klar, testa den i en validator för att få hjälp att upptäcka syntaxfel.

HTML: https://validator.w3.org/
CSS: https://jigsaw.w3.org/css-validator/

3. Om allt stämmer enligt validatorerna, har vi nu en enkel webbsida som inn

 Om allt stämmer enligt validatorerna, har vi nu en enkel webbsida som innehåller ett utbud av vanliga element, en del CSS-finesser samt JS som ändrar både innehåll (HTML) samt utseende (CSS).

```
HTML-koden:
```

```
<!doctype html>
<html>
      <head>
            <meta charset="utf-8">
            <link href="style.css" media="screen" rel="stylesheet"</pre>
                 type="text/css"/>
            <script src="lektion2.js"></script>
            <title>Lektion 2</title>
      </head>
      <body>
            <div id="left" class="content"</pre>
                 onclick="changeColor(this.id)"></div>
            <div id="mid" class="content"</pre>
                 onclick="changeColor(this.id)"></div>
            <div id="right" class="content"</pre>
                 onclick="changeColor(this.id)"></div>
            <div id="textBox" class="content"</pre>
                 onclick="resetPage()">
                 0
            </div>
      </body>
</html>
CSS-koden:
body {
     background-color: #6a6a6a;
}
.content {
    float: left;
    width: 20vw;
    height: 20vw;
    margin: 2vw;
    border-radius: 2vw;
    border: solid 2px #fff;
    box-shadow: 0 6px 12px 3px rgba(0, 0, 0, 0.4);
    cursor: pointer;
    -webkit-transition: background-color 0.2s ease-in-out;
```

```
-moz-transition: background-color 0.2s ease-in-out;
    -o-transition: background-color 0.2s ease-in-out;
    -khtml-transition: background-color 0.2s ease-in-out;
    transition: background-color 0.2s ease-in-out;
}
#textBox {
      clear: left;
      margin: 2vw 26vw;
      line-height: 5vw;
}
p {
      font-family: 'Open Sans Condensed', sans-serif;
      font-size: 8vw;
      font-weight: 300;
      color: #ff7d00;
      text-align: center;
}
#left {
      background-color: #a6a6a6;
}
#mid {
      background-color: #ff7d00;
}
#right {
      background-color: #a6a6a6;
}
JS-koden:
function changeColor(id){
      var numberOfClicks =
            parseInt(document.getElementById("clickCounter").textContent);
      document.getElementById("clickCounter").textContent =
            numberOfClicks+1;
      if (id == "left") {
      document.getElementById("left").style.backgroundColor = "#ff7d00";
      document.getElementById("mid").style.backgroundColor = "#a6a6a6";
      document.getElementById("right").style.backgroundColor = "#a6a6a6";
      } else if (id == "mid") {
      document.getElementById("left").style.backgroundColor = "#a6a6a6";
      document.getElementById("mid").style.backgroundColor = "#ff7d00";
      document.getElementById("right").style.backgroundColor = "#a6a6a6";
      } else {
      document.getElementById("left").style.backgroundColor = "#a6a6a6";
      document.getElementById("mid").style.backgroundColor = "#a6a6a6";
      document.getElementById("right").style.backgroundColor = "#ff7d00";
      }
}
function resetPage(){
      document.getElementById("clickCounter").textContent = 0;
      document.getElementById("left").style.backgroundColor = "#a6a6a6";
document.getElementById("mid").style.backgroundColor = "#ff7d00";
      document.getElementById("right").style.backgroundColor = "#a6a6a6";
}
```