Redogörelse Individuellt Projekt

- HTML5 och CSS3 (HTML & CSS 1.1)
- Responsiv design (HTML & CSS 1.2)
- SEO/Sökmotoroptimering (HTML & CSS 1.3)
- Tillgänglighetsanpassning (HTML & CSS 1.4)
- JavaScript och hur det i grunden fungerar/är uppbyggt (Javascript 1.1)
- JavaScriptsversioner och dess kompabilitet (Javascript 1.2)
- Hur JavaSkript körs i webbläsaren (Javascript 1.3)
- DOM (Javascript 1.6)
- Utvecklingsverktyg du använt (Arbetsmetodik 1.1)

HTML5 och CSS3 (HTML & CSS 1.1)

HTML och CSS är grunden i en hemsida. HTML är själva ritningen som beskriver hur koden är strukturerad semantiskt. CSS används för att styla HTMLelementen som är byggstenarna. Med CSS kan du flytta runt de olika elementen till önskvärd plats, byta storlek och typsnitt på rubriker och text och responsivt designa din hemsida så att den är kompatibel med andra devices.

Med det sagt är större delen av mitt projekt är uppbyggt på HTML och CSS. I HTMLdokumenten har jag angett vilka element jag vill ska befinna sig på hemsidan och nästan allt visuellt har designats med CSS.

Responsiv design (HTML & CSS 1.2)

Responsiv design är en stor del av att utveckla en hemsida. Det är omöjligt att veta vilken typ av device och skärmstorlek dina besökare kommer att använda. En dåligt designad hemsida kan se jättebra ut när du besöker den på en dator men bli väldigt rörig när du besöker den med en smartphone. Element kan krocka och smälta ihop med varandra, bilder och text kan hamna på ett helt annat ställe än vad du tänkt dig. Detta kan undvikas om man tänker ett steg längre. Det finns nåra småknep och tricks jag har använt mig av för att undvika detta och göra min hemsida presentabel oavsett vilken skärmstorlek den besöks med.

Den mest grundläggande metoden för responsiv design är Media Queries. En Media Query tillåter dig att köra ett block av CSS-kod om ett speciellt utsett condition är true. I mitt fall reagerar mina Media Queries på pixelstorleken av besökarens skärm/display och tillåter hemsidan att anpassa sig storleksmässigt med önskvärd design för just denna storlek.

Andra knep jag har använt mig av är att inte sätta storlek på saker och ting med pixlar. Man kan istället ange storlek i % eller em. På så sätt blir sidan nedskalningsbar.

SEO/Sökmotoroptimering (HTML & CSS 1.3)

Sökmotorer som Google och Bing läser av din hemsida för att uppskatta relevansen av det användaren har sökt på. Då är det viktigt att man har byggt upp sin sida med god semantik och relevant innehåll. Om jag ser till mitt egna projekt är detta delen som jag funnit mest problematisk då jag tycker att min hemsida är semantiskt uppbyggd samtidigt som den inte dyker upp om jag googlar på den.

Tillgänglighetsanpassning (HTML & CSS 1.4)

En bra hemsida skall vara anpassad till en bred publik, även folk som lider av eventuella handikapp så som synskador. För att en synskadad ska kunna uppfatta, förstå och navigera sig på hemsidan är det viktigt att den har en semantisk struktur och att du använder rätt element för rätt innehåll, annars är det väldigt svårt att uppfatta vart på sidan man befinner sig. Allt som ligger inom en meny skall ligga inom speciellt utsedda taggar, i detta fall <nav></nav> och en footers innehåll ska ligga inom <footer></footer>. Ett element däremot kan vara svårt att uppfatta då det inte är ett specifierat element så innehållet kan variera kraftigt. När en synskadad besöker en sida får de elementen upplästa för sig och på så sätt kan skapa en bild av vart på sidan de befinner sig.

En bra designad hemsida skall inte enbart vara anpassade till människor med handikapp utan även äldre människor ska kunna navigera sig fram utan problem. Därför har jag hållt mig till en relativt simpel styling med tydliga rubriker och sett till att sidan inte består av massa onödig information som kan ta fokus från det viktiga innehållet.

Javascript och hur det i grunden fungerar/är uppbyggt (Javascript 1.1)

Javascript är grundepelaren till det mesta av funktionaliteten på en hemsida och har många anvädningsområden. Det kan vara allt ifrån en ett rullande bildspel till att kontrollera ifyllda fält i ett formulär innan det skickas till en server. Denna kod kan vara skriven inom <script></script>-taggar direkt i HTMLdokumentet eller ligga externt i en .js-fil som du sedan kopplar ihop med ditt HTMLdokument. Jag har använt mig av båda dessa metoder i mitt projekt. Fördelen med att ha javascript-koden för sig på en egen fil är att HTMLdokumentet inte blir lika rörigt samt att om du skall använda samma kod för flera sidor slipper du klippa in kodstycket i samtliga HTMLdokument. Men när det handlar om ett mindre kodstycke som enbart skall användas på en sida kan det vara skönt att ha den i direkt i HTMLdokumentet för att enkelt kunna ändra om.

Jag har knytit en .js-fil med namnet countdown2.js till min förstasida, vilket är en nedräknare till examen samt LIA. Denna funktion kräver en viss logik bakom sig och javascriptet är "hjärnan" som tolkar koden samt infogar resultatet inom ett utsatt HTML-element för att sedan kunna visas i din webbläsare när du öppnar sidan.

JavaScriptsversioner och dess kompabilitet (Javascript 1.2)

Olika browsrar kör olika versioner av javascript. Även fast ditt script körs utan problem i Google Chrome kan samma script ställa till problem om du öppnar sidan i Internet Explorer då dom stöder olika versioner av javascript. Det är omöjligt i förhand att veta vilken browser dina besökare kommer att använda sig av men man kan helgardera sig genom att använda <noscript>-taggen som tillåter dig att ange alternativt innehåll om browsern i fråga inte kan köra ditt script.

Hur Javascript körs i webbläsaren (Javascript 1.3)

För att implementera din javascript-kod i hemsidan måste man lägga koden inom <script></script>-taggar alternativt i en extern .jsfil. Sedan skapar du en container i ditt HTMLdokument med ett ID eller en Class för att kunna peka ut vart ditt javascript skall köras. I min countdown.js finns det två separata funktioner som anger nedräkningens deadline. När nedräknarna har all bakomliggande logik för att fungera som den ska har jag valt ut vart i HTMLdokumentet jag vill att dessa nedräkningar skall visas. document.getElementById("clockdiv").innerHTML = kvar; pekar ut att jag vill visa nedräknaren inom ID:t "clockdiv" som är ett -element i min HTMLkod.

DOM (Javascript 1.6)

DOM står för Document Object Model och brukar oftast beskrivas som ett träd i sin struktur. Med DOM kan javascript komma åt och ändra alla elementen av ett HTML-dokument. Ett exempel på detta är kodstycket jag beskrev ovanför.

Utvecklingsverktyg du använd (Arbetsmetodik 1.1)

Den allra största delen av utvecklingen av mitt projekt har skett i editorn Sublime Text. I början använde jag mig av Visual Studios men bytte då jag upplevde den editorn som väldigt prestanda-krävande och tung för datorn att köra. Sublime Text är mer nedskalad och tar inte längre tid att starta upp än ett vanligt textdokument.

För att ladda upp mitt projekt på domänen använde jag mig av FileZilla som är en FTP.

Jag har även använt mig av dropbox för att backa upp mina filer ifall hårddisken eller dylikt skulle krascha så att jag slipper börja om från början om något sådant skulle ske.

Trello har hjälpt mig att få en klarare överblick av mitt projekt och gjort det enklare för mig att planera processen.