

HTML-Dokument

- Att titta på en sida direkt i browsern säger inget om strukturen
- Inspect element i chrome ger oss tillgång till strukturen. Muspekaren över strukturen markerar i webbläsaren
- HTML-taggar är som mackor, två bröd med innehållet mellan.
- I head har vi allt som får mackan att funka men det behövs inte. Icke visuella element som ger funktion åt hemsidan.
- I body har vi köttet och grönsakerna, det som gör mackan. Body innehåller all visuellt innehåll på hemsidan.
- Nedanför följer alla taggar för sig

Doctype

- Historien om doctype är lång
- Doctype är något som få vet mer om än att de behöver den på hemsidan
- Det doctype egentligen gör är att inte trigga "quirks mode" vilket säkerställer rendering på rätt sätt.
- Du behöver den och den ska spegla den version av html du har skrivit på hemsidan.
- Anledning till att vi tittar på doctype när den ändå har en sådan simpel funktion är för att många av er kommer att arbeta med gammal kod i framtiden. Detta innebär att det då och då kommer att dyka upp en konstigt doctype och "OMG vad är all denna kod!?".
- När vi bygger en sida börjar vi med `<!doctype HTML>`.

Head

- Här ligger all information som useragenten måste veta för att hantera sidan på rätt sätt. Detta inkluderar referenser till andra filer och metataggar.
- Innan vi deklarerar head måste vi ha HTML taggarna, de yttersta brödbitarna som omfamnar vår hemsida.
- I html-taggen kan det vara bra att lägga till ett lang attribut för att berätta vilket språk innehållet är skrivet på för den useragent som besöker sidan. Det är nice men är inte ett krav
- I head ska vi lägga metataggar för att bland annat:
 - Berätta vilken character encoding vi har på sidan, utf-8 är en bra standard om vi inte skriver på ryska eller asiatiska språk.
 - beskrivning av sidans innehåll med `name="description"` och `content=""`
- Vi lägger även in title i head samt eventuella länkar till externa filer så som CSS och ibland Script.

Body

- Head innehåller allt funktionell information som är i stort sett osynlig
- Body innehåller allt det synliga innehåll som hemsidan innehåller
- Body är en "sibling" till head vilket betyder att de ligger på samma nivå i hierarkin.
- Body behövs för att ha synlig content på hemsidan. Läger vi in lite text så får vi en typ av placeholder som ger detta content browsern default-style.
- Läger vi till ett element så som h1 får vi en huvudrubrik.
- Läger vi till ett element så som p får vi en paragraf.

Content models

- Alla element i HTML tillhör minst en content model som kontrollerar syntax så som vilka element som kan vara nästlade inom varandra.
- Innan HTML5 fanns det i stort sett två stycken content models, inline och block.
- Block: Tar upp en egen rad i "content flow"
- Inline: Ligger i "content flow"
- HTML5 har 7 content models
- Metadata – sätta upp presentation och beteende
- Embedded – importerar material från andra källor
- Interactive – för användarinteraktion
- Heading – rubriksättning
- Phrasing – Inline
- Flow – majoriteten av elementen. Element som normalt lägger sig i "flowet"
- Sectioning – Skapar nya sektioner inom dokumentet
- <http://www.w3.org/TR/html5/dom.html#content-models>

Formatering av content med HTML

- HTML-taggar används för att berätta vad innehållet är
- Pre – elementet kan användas för att behålla innehållet som det är
- H – används för rubriksättning, användes felaktigt för om man ville ändra storlek på text. Detta sköts av css.
- Finns taggar för att ändra presentationen av text, så kallad presentational tags
- <i> och ändrar presentationen till kursivt respektive fetstilt.
- Det finns taggar som gör samma sak men lägger till mening, så kallade logical tags
- och ändrar presentationen kursivt respektive fetstil. Ändrar även hur useragent läser av materialet och ger mening till innehållet. Använd dessa när du vill ge mening till content.

Rubriker

- Hierarkin visas med hjälp av textstorlek
- Vi vill utveckla en strategi för hur vi ska rubrikätta materialet
- Om allt får en h1-taggar har alla samma "importance". Välj rubriknivå utifrån hur viktiga rubrikerna är.
- Med extensions till en texteditor kan man se hur document-outline ser ut. Blir strukturerat som ett word-dokument.
- Hoppa ej över rubriknivåer

Paragrafer

- Den mest basic formateringen i html
- Browsern tar för givet att saker vi glömmer att formatera själva är paragrafer. Språket är väldigt förlåtande.
- Sparar oss tid och inga buggar dyker upp men kan ha den direkt motsatta effekten
- Ibland tror vi att någonting händer när det egentligen inte gör det
- Paragraftaggen ska användas för alla individuella paragrafer och inte för att separera block av text helt villt.
- Paragrafer har margin upp och ner som ej kollapsar med andra paragrafers margin

Kontrollera line-breaks

- Ibland måste man skapa en radbrytning men inte vill använda en ny paragraf för det. Istället använder vi line-breaks
- Om vi har en adress inom en p-tagga ska vi inte bryta ut de separata raderna inom egna p-taggar då det blir semantiskt inkorrekt.
- `
` hade förut `
`, detta är XHTML
- Linebreaks skapar inte nya sektioner i koden

Betona text

- Ibland vill man betona content på vår sida
- Bold
- Strong
- Italic
- Emphasize
- Ibland vill man markera text utan att betona
- Xhtml tog bort b och i men de kom tillbaka i html5.

Speciella karaktärer och tecken

- Ibland vill man använda tecken och symboler som inte finns på tangentbordet eller som redan är reserverade i html.
- Vi använder named character entity
- `<` blir `&lang`
- `>` bli `&rang`
- Användning av någon av dessa på sitt normala sätt, `"<"`, skapar kaos i koden.
- Copyright och trademark finns inte på tangentbordet, finns som named character entity