1.2 Webbutvecklingens miljöer

> Webbutveckling är i allra högsta grad ett praktiskt ämne.

Man kan inte lära sig webbutveckling genom att endast läsa böcker. För att lära sig webbutveckling måste man skriva kod och testa koden – precis som när man lär sig att köra bil. Och för att göra det behöver man en miljö, där man kan skriva och en annan där man kan köra och testa koden. Den första är en texteditor, den andra en webbläsare (browser). Det finns en uppsjö av utvecklingsmiljöer för de olika språken. Ofta kallas de för IDE, *Integrated Development Environment* – nästan för avancerade för oss. En enklare variant – ofta en del av en IDE – är en *editor*.

Editorer

En *editor* är ett skrivverktyg på datorn, dvs ett program som kan hantera text. *Ordbehandlingsprogram* är en annan beteckning på editorer. På de flesta datorerna finns minst en editor förinstallerad. För att skriva kod och spara den i en fil behövs en editor. Men kod får innehålla endast sådana tecken som kan 'förstås' av webbutvecklingsspråket. Därför måste editorn spara filen som *oformaterad textfil*, dvs utan styr- och kontrollkoder som i vissa ordbehandlingsprogram används för formatering (typsnitt, stil, sorlek osv.). Ett exempel på sådana program är Word som formaterar texten och sparar sina filer som dokument av typ **`*.docx`**. Formatering innebär att det läggs till osynliga tecken i texten som webbutvecklingsspråket inte känner till. Motsvarigheten på Mac-datorer är Pages. Sådana ordbehandlingsprogram är **inte** lämpliga för att skriva kod. Däremot kan t.ex. Notepad (Anteckningar), Notepad++ eller TextPad på Windwos-datorer och Textredigerare, TextEdit eller Emacs på Mac-datorer vara lämpliga texteditorer för webbutveckling och programmering, eftersom de sparar alla filer som rena textfiler *utan* formatering. För textfiler kan filändelser av typ **`*.txt`** väljas, men även andra, beroende på operativsystemet.

IDE står för *Integrated Development Environment*, är alltså en integrerad programutvecklingsmiljö som inkluderar en editor, en *interpretator* resp. *kompilator* och andra verktyg för programutveckling i en och samma samlad miljö. *Visual Studio* är ett exempel på en IDE som har utvecklingsverktyg för ett antal språk. Men JavaScript behöver ingen IDE. Det räcker med en texteditor där koden skrivs och sparas samt en webbläsare där koden exekveras. Vi kommer att använda oss av denna möjlighet som är oberoende av tredje parts verktyg för att slippa installera nya program.

Interpretator vs. kompilator

En *interpretator* är ett program som *tolkar* källkod till maskinkod och skickar maskinkoden till datorns processor utan att mellanlagra den på hårddisken. Processorn exekverar maskinkoden. Källkod är kod som endast människan förstår, men inte

datorn. Maskinkod är kod som endast datorn förstår, men inte människan. Alla webbutvecklingsspråk är interpreterande, inkl. HTML.

Till skillnad från en interpretator är en *kompilator* ett program som *översätter* källkod till maskinkod och lagrar maskinkoden på hårddisken. Först när man exekverar skickas den kompilerade maskinkoden till datorns processor och utförs där. Vissa programmeringsspråk är kompilerande (C/C++), andra är interpreterande (Python).

Webbläsare

En *webbläsare*, på engl. *web browser* är ett program som kan *tolka* HTML-kod samt andra scriptspråkens koder. Att *tolka* är synonym till att *interpretera*, *exekvera* eller *utföra*. Webbläsaren är alltså i huvudsak en HTML-interpretator. Själva programmet har skrivits i något av de universella språken, ofta i C. Exempel på sådana program är Google Chrome, Internet Explorer, Firefox, Safari, Netscape, ... En webbläsare finns förinstallerad på alla datorer.

HTML-kod kan exekveras av webbläsaren både lokalt och från Internet. I båda fall måste koden vara sparad i en fil. Ska HTML-koden exekveras i en webbläsare måste filen som innehåller koden, ha ändelsen **`html`**, vare sig vi har lagrat filen lokalt eller hämtat den från en server på Internet. När vi testar våra koder i webbutvecklingskursen gör vi det lokalt.

HTML – webbens standardspråk

HTML står för ***H**yper**T**ext **M**arkup **L**anguage* och är webbens standardspråk. Syftet med HTML är att producera presentabla *dokument* som kombinerar text, bild och andra element. Koden genererar dokumentet som sedan visas på webben. Koden man skriver, är separerad från dokumentet – till skillnad från andra formateringsverktyg som t.ex. *Word*, där man direkt skriver i dokumentet. Man talar om att Word är ett s.k. ***WYSIWYG***-verktyg, dvs ***W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et*. I Word *ser* man, vad man *får* i dokumentet. Koden (VB Script) genereras automatiskt i bakgrunden och är osynlig för användaren.

Till skillnad från Word är HTML ett icke-*WYSIWYG*-verktyg, dvs man skriver koden utan att se dokumentet. Koden måste först tolkas av en interpretator, innan dokumentet kan uppstå. Webbläsaren som själv en programvara, är en sådan interpretator som exekverar koden och visar dokumentet. Webbläsaren har utvecklats för att bl.a. tolka HTML. Andra exempel på icke-*WYSIWYG*-verktyg är TeX/LaTeX, en applikation för typsättning och presentation av text, speciellt matematiska uppsatser.

En viktig egenskap av HTML är:

> HTML är **inte** case sensitive (skiftlägeskänslig).

Dvs HTML skiljer inte på små och stora bokstäver.

Script vs. program

Med *script* menas all kod som, inbäddad i HTML, kan köras på webben inkl. HTML-kod själv. Och språk som används i en sådan kod kallas för *scriptspråk*. Exempel på scriptspråk är *JavaScript*, *PHP*, ... JavaScript används i regel på klientdatorer, medan PHP är webbservrarnas scriptspråk. Koder till *universella* språk, som kan användas för vilken applikation som helst och inte är begränsade till webben, kallas för *program*.

Det finns två olika kategorier av språk i datavärlden: *scriptspråk* och *universella* språk som t.ex. C, C++, C#, Java, Python, ... HTML är ett *scriptspråk* (Läs om script i nästa paragraf). HTML har även möjligheten att bädda in andra scriptspråk i sin kod som t.ex. JavaScript, förutsatt att man avgränsar språken genom tydliga markeringar. Webbläsaren är den naturliga exekveringsmiljön både för HTML och alla andra scriptspråk, medan särskilda verktyg behövs för att exekvera universella språk. För att *skriva* scriptkod behöver man endast en *editor*, och för att exekvera den en *webbläsare*. Vi nöjer oss med denna minimalistiska miljö vad gäller scriptspråken, för att förenkla den tekniska hanteringen och koncentrera oss på själva språket.

Det mest kända scriptspråket är *JavaScript* som skapades år 1995 av *Netscape*, ett amerikanskt mjukvaruföretag som 1994 lanserade den första grafiska webbläsaren *Mosaic* som snabbt blev en jättesuccée. Netscape integrerade JavaScript i Mosaic, för att göra webben interaktiv, se ***De tre skikten*** sid 7.

JavaScript får inte förväxlas med Java. Det handlar om två olika programmeringsspråk som dessutom tillhör två olika kategorier av programmeringsspråk: Medan JavaScript är ett scriptspråk är Java ett universellt programmeringsspråk.

Hos scriptspråken nöjer man sig med de enklare elementen i programmering, för att förse webbsidor med vissa funktionaliteter. Scripten bakas in i HTML-kod, varför de kan exekveras på webben.

> Scriptspråkens *exekverings*miljö är webbläsaren (web browser).

Scriptspråken är *interpreterande* språk, dvs koden tolkas till maskinkod (datorns språk) av en interpretator som är inbyggd i webbläsaren. Maskinkoden utförs direkt av datorns processor utan att den mellanlagras. De mest använda webbläsarna är Google Chrome på Windwos-datorer och Safari på Mac-datorer. I båda är en interpretator för JavaScript inbyggd.

*Utvecklings*miljön däremot – dvs där man skriver koden – kan vara vilken editor som helst. Till skillnad från HTML är JavaScript-kod case sensitive, dvs JavaScript skiljer på små och stora bokstäver. Det gör inte HTML.

Filändelser

Skriver du din JavaScript kod i någon editor och sparar filen som **`*.txt`**, kommer du inte kunna exekvera den i en webbläsare, när du (dubbel)klickar på den. Boven i dramat är filändelsen: Operativsystemet identifierar de filer som innehåller kod via filändelsen. All JavaScript kod är inbakad i HTML kod, webbläsarens språk. Ska koden exekveras i en webbläsare måste filen som innehåller koden, ha ändelsen **`html`**, för att kunna identifieras som en JavaScript källkodsfil. Därför måste du antingen från början spara din källkodsfil med ändelsen **`html`** eller i efterhand ändra filändelsen till **`html`**. I Windows kallas filändelser för *Filnamnstillägg*.

För att kunna följa reglerna för filändelsen som beskrevs ovan, förutsätts att man kan *se* filändelserna när man öppnar en mapp. Men i praktiken är detta ofta inte fallet. Orsaken år på operativsystemets inställningar. I Windows är default inställningen att man i regel *inte* kan se dem. Ta själv reda på hur det är på din dator. Så här kan man göra för att synliggöra filändelserna i Windows:

- Öppna en mapp i Windows.
- Gå i mappens menyrad till Mappalternativ. Om du inte hittar denna meny klicka på de tre små punkterna till höger (Visa mer) och välj Alternativ.
- Du borde få upp dialogrutan Mappalternativ. Välj fliken Visning. Bocka av rutan Dölj filnamnstillägg för ända filtyper. Så här borde nu dialogrutan se ut:

- Klicka på knappen Använd i alla mappar, sedan på Ja och OK.

Nu borde du kunna se dina filers ändelser och kunna följa reglerna på förra sidan. Generellt rekommenderas att ha synliga filändelser på sin dator, när man programmerar.