

# En praktisk introduktion till webbutveckling med Drupal och WordPress

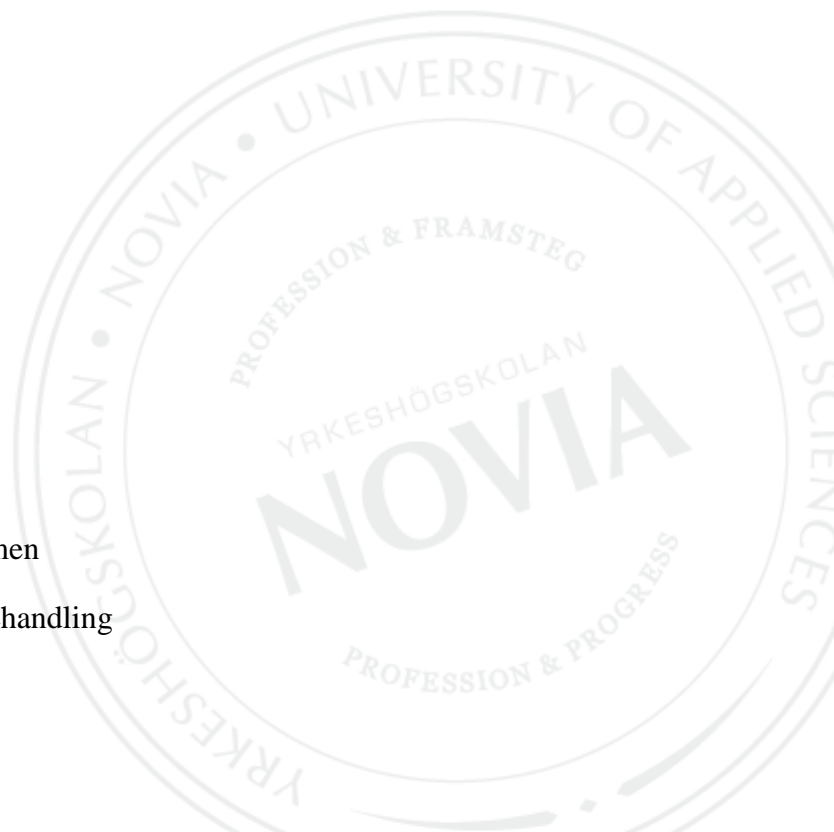
Tomas Söderlund

Johan Österberg

Examensarbete för Tradenom (YH)-examen

Utbildningsprogrammet i Informationsbehandling

Raseborg 2014



## **EXAMENSARBETE**

Författare: Tomas Söderlund och Johan Österberg

Utbildningsprogram och ort: Tradenom, Raseborg

Inriktning/alternativ/Fördjupning: Informationsbehandling

Handledare: Rolf Gammals

**Titel: En praktisk introduktion till webbutveckling med Drupal och WordPress**

---

Datum: 14.4.2014

Sidantal: 62

Bilagor: 0

---

### **Sammanfattning**

Examensarbetet fungerar som en introduktion till webbutveckling med hjälp av innehållshanteringssystemen Drupal och WordPress. Inom ramen för detta examensarbete har vi skapat två identiska webbplatser med Drupal respektive WordPress. I arbetet jämför vi funktionaliteten och utvecklarvänligheten på dessa webbplatser. För att det skall vara så nybörjarvänligt som möjligt beskrivs arbetsprocessen mycket ingående.

Webbplatsen skall fungera responsivt eftersom det är en grundläggande funktion i webbutveckling i dagens läge. Vi går noggrant in på hur det här fungerar i praktiken.

Eftersom vi jobbade med de båda CMS-systemen Drupal och WordPress gjorde vi en jämförelse av de båda systemen. Detta gör vi genom slutsatser vi drar från det praktiska arbetet och från intervjuer med experter. Vi har intervjuat en Drupalexpert och en WordPress expert.

Resultatet av arbetet är att vi skapat en responsiv webbplats där arbetsprocessen fungerar som en introduktion till webbutveckling och CMS-systemen Drupal och WordPress. Vi har också gjort en klar och tydlig jämförelse av de här två systemen.

---

Språk: Svenska    Nyckelord: innehållshanteringssystem, responsiv, webbutveckling

---

## **OPINNÄYTETYÖ**

Tekijä: Tomas Söderlund ja Johan Österberg

Koulutusohjelma ja paikkakunta: Tradenomi, Raasepori

Suuntautumisvaihtoehto/Syventävät opinnot: Tietojenkäsittely

Ohjaaja: Rolf Gammals

### **Nimike: Käytännön johdatus web-kehitykseen Drupalin ja WordPressin avulla**

---

Päivämäärä: 14.4.2014

Sivumäärä: 62

Liitteet: 0

---

#### **Tiivistelmä**

Opinnäytetyö toimii johdantona web-kehitykseen sisällönhallintajärjestelmien Drupalin ja WordPressin avulla. Tämä tarkoittaa käytännössä sitä että luomme kaksi samanlaista sivustoja Drupalin ja WordPressin avulla jossa vertaamme toiminnallisuutta ja käytettävyyttä. Jotta opinnäytetyö olisi niin aloittelijaystävällinen kuin mahdollista kuvaamme työprosessia hyvin yksityiskohtaisesti.

Sivusto toimii Responsive web design periaatteella koska se on olennainen osa web-kehittelyä tällä hetkellä. Selitämme mitä tämä tarkoittaa ja miten se käytännössä toimii perusteellisesti.

Työskennelyämme CMS-järjestelmien Drupalin ja WordPressin kanssa teimme vertailun järjestelmien kesken. Tämän teimme vetämällä johtopäätöksiä työprosessista ja haastatteluista asiantuntijoiden kanssa. Olemme haastattelleet Drupal asiantuntijan ja WordPress asiantuntijan.

Työn tulos on responsiivinen sivusto jossa työprosessi toimii johdantona web-kehitykseen ja CMS-järjestelmiin Drupal ja WordPress. Olemme myös tehneet selvän vertailun näiden kahden järjestelmien kesken.

---

Kieli: Ruotsi

Avainsanat: sisällönhallintajärjestelmä, responsiivinen, web-kehittely

---

## **BACHELOR'S THESIS**

Author: Tomas Söderlund and Johan Österberg

Degree Programme: Business Information Technology, Raseborg

Specialization: Information processing

Supervisors: Rolf Gammals

**Title: A practical introduction to web development through the two Content Management Systems Drupal and WordPress**

---

Date: 14.4.2014

Number of pages: 62

Appendices: 0

---

### **Summary**

The thesis functions as an introduction to web development by using the Content Management Systems Drupal and WordPress. In practice we have created two identical websites using Drupal and WordPress in which we compare the functionality and how user-friendly the systems are. To be as beginner-friendly as possible we describe the work-process very thoroughly.

The website will be responsive because that is a basic function today in web development. We describe how to make the website responsive in practice.

As we worked with the two CMS systems Drupal and WordPress we did a comparison of the two systems. We did this by drawing conclusions from the practical work we did and by interviewing experts. We have interviewed an expert on Drupal and an expert on WordPress.

The result of the thesis is that we have created a responsive website where the work process serves as an introduction to web development and the CMS systems Drupal and WordPress. We also made a clear and concise comparison between these two systems.

---

Language: Swedish

Key words: content management system, responsive, web development

---

## Innehåll

|       |  |    |
|-------|--|----|
| 1     | Inledning.....                         | 1  |
| 1.1   | Mål.....                               | 1  |
| 1.2   | Disposition.....                       | 2  |
| 1.3   | Målgrupp.....                          | 3  |
| 2     | Val av verktyg.....                    | 3  |
| 2.1   | HTML och CSS.....                      | 4  |
| 2.2   | PHP.....                               | 5  |
| 2.3   | MySQL.....                             | 6  |
| 2.4   | Notepad++ och Firebug.....             | 6  |
| 3     | Planering och indelning av arbete..... | 7  |
| 4     | CMS-system.....                        | 7  |
| 4.1   | Insticksmoduler.....                   | 8  |
| 4.2   | Drupal.....                            | 8  |
| 4.2.1 | The Drupal Way.....                    | 9  |
| 4.3   | WordPress.....                         | 10 |
| 4.3.1 | WordPress.com.....                     | 10 |
| 4.3.2 | Automattic Ink.....                    | 11 |
| 5     | Design.....                            | 11 |
| 5.1   | Responsive Web Design.....             | 12 |
| 5.1.1 | Responsiv design i praktiken.....      | 12 |
| 6     | Installation.....                      | 15 |
| 7     | Tema.....                              | 20 |
| 7.1   | Tema i Drupal.....                     | 20 |
| 7.2   | Tema i WordPress.....                  | 21 |
| 8     | Installera moduler i Drupal.....       | 22 |

|        |   |    |
|--------|---|----|
| 8.1    | Installera moduler i WordPress.....     | 22 |
| 9      | Block/Widget.....                       | 23 |
| 9.1    | Block i Drupal.....                     | 23 |
| 9.2    | Widget i WordPress .....                | 23 |
| 10     | Navigationsmeny .....                   | 24 |
| 10.1   | Navigation i Drupal.....                | 25 |
| 10.1.1 | Responsiv navigation.....               | 26 |
| 10.2   | Navigation i WordPress .....            | 27 |
| 10.2.1 | Responsiv navigation.....               | 28 |
| 11     | Flerspråkig webbplats.....              | 30 |
| 11.1   | Flerspråkig webbplats i Drupal .....    | 30 |
| 11.2   | Språkval i WordPress .....              | 34 |
| 12     | Mera funktionalitet till noderna.....   | 36 |
| 12.1   | Panels i Drupal .....                   | 36 |
| 12.2   | Advanced Custom Field i WordPress ..... | 40 |
| 13     | Senaste nyheter .....                   | 43 |
| 13.1   | Senaste nyheter i Drupal .....          | 43 |
| 13.2   | Senaste nyheter i WordPress .....       | 48 |
| 14     | Galleri.....                            | 49 |
| 14.1   | Galleri i Drupal.....                   | 50 |
| 14.2   | Galleri i WordPress .....               | 51 |
| 15     | Intervju .....                          | 52 |
| 15.1   | Kvalitativ metod.....                   | 52 |
| 15.2   | Egenföretagere Mathias Bilenberg.....   | 53 |
| 15.3   | Intervju med Mathias Bilenberg.....     | 53 |
| 15.4   | Codes Digital Media.....                | 55 |

|      |   |    |
|------|---|----|
| 15.5 | Intervju med Codes Digital Medias grundare Conny Karlsson ..... | 55 |
| 16   | Skillnader mellan Drupal och WordPress .....                    | 57 |
| 17   | Utvecklingsmöjligheter .....                                    | 60 |
| 18   | Slutsatser.....   | 60 |
| 19   | Avslutning .....  | 61 |
|      | KÄLLFÖRTECKNING .....   | 63 |
|      | FIGURFÖRTECKNING .....  | 64 |
|      | KODFÖRTECKNING .....  | 66 |
|      | TABELLFÖRTECKNING.....  | 67 |

# 1 Inledning

Tanken bakom arbetet är att se skillnaderna mellan de två olika cms-systemen Drupal och WordPress. Vi kom på idén själva eftersom vi har erfarenhet av båda systemen och vi tyckte att det skulle vara intressant att göra en ordentlig jämförelse för att ta reda på vilka skillnader som finns i utvecklandet av en webbplats. Eftersom vi startade arbetet på eget initiativ så saknar vi uppdragsgivare.

Idén kom i och med att Tomas Söderlund jobbade med WordPress under praktiken hos företaget CODES digital media som är specialiserat bara på WordPress. Vi har båda erfarenhet i att använda Drupal men Johan Österberg är den som har arbetat mera med detta system.

För att se skillnaderna kommer det i praktiken att ske så att Johan Österberg skapar en webbplats med hjälp av Drupal och Tomas Söderlund skapar en identisk webbplats med hjälp av WordPress. Designen på webbplatsen kommer att vara ganska enkel eftersom det i praktiken görs på samma sätt i båda CMS-systemen. Fokuset kommer därför att ligga på funktionaliteten och uppbyggandet av webbplatsen.

## 1.1 Mål

Det finns tre huvudmål i arbetet:

1. En praktisk introduktion till webbutveckling.
2. Responsiv design.
3. Skillnader mellan CMS-systemen Drupal och WordPress.

Genom att bygga upp en webbplats från grunden med hjälp av Drupal och Wordpress kommer vi att göra en praktisk introduktion till webbutveckling. Den kommer att vara så nybörjarvänlig som möjligt vilket betyder att arbetet kan fungera som en grund för att börja med webbutveckling.



Ett viktigt mål för oss är att webbplatsen skall vara responsiv eftersom vi anser att det är någonting alla webbutvecklare måste kunna i dagens läge. Det här betyder att grundlayouten och alla moduler är implementerade så att de fungerar responsivt.

Eftersom den praktiska introduktionen görs med hjälp av Drupal och WordPress kommer arbetet också att visa skillnader mellan de två systemen. Det här kommer vi att göra genom att jämföra arbetsprocessen och dra slutsatser från det. Vi kommer också att granska statistik och jämföra systemen så. Slutligen kommer vi att göra intervjuer med personer som har erfarenhet av att jobba med CMS-systemen i arbetslivet p.g.a. att vi vill ha en inblick av professionella webbutvecklare. Den här jämförelsen gör vi med den kvalitativa forskningsmetoden vilket vi tar upp i samband med intervjuerna.

Det viktigaste målet för oss var att jämföra uppbyggnaden av en vanlig webbplats med CMS-systemen Drupal och WordPress. Det var också viktigt för oss att utveckla vårt kunnande om responsiv webbdesign.

## 1.2 Disposition

Arbetet har de här delarna:

- Design
  - Responsiv design
- Funktionalitet (insticksmoduler)
- Intervju med två specialister på Drupal och WordPress
- Skillnader och slutsatser

Vi förklarar nu kort alla de här punkterna.

Designen är inte vår högsta prioritet men eftersom det är så viktigt i dagens samhälle där så många använder olika elektroniska apparater som pekplattor och smarttelefoner för att surfa på nätet, så har vi beslutit oss för att använda responsiv design.

Båda webbplatserna skall ha samma funktionalitet trots att vi kommer att använda olika insticksmoduler på grund av att det finns skillnader i de olika systemen. De olika delarna är som följer:

- Dropdown navigation
- Galleri
- Responsiv design
- Nyhetsida och nyhetsbalk
- Flera språk

Vi går mera in på djupet på de här olika delarna när vi beskriver arbetsprocessen.

Arbetsprocessen kommer att förklaras på ett sätt som lämpar sig för personer utan tidigare erfarenhet av att använda CMS-system.

För att få ett perspektiv från arbetslivet har vi valt att intervjua Conny Karlsson, från CODES digital media, som har gått över till WordPress från sitt eget kodade CMS-system. Vi har också intervjuat Mathias Bilenberg för att få ett perspektiv av en Drupalanvändare.

Till slut kommer vi att göra en jämförelse av de olika systemen vilket innebär att vi kollar vilka skillnader som fanns i arbetsprocessen och drar slutsatser från det. Vi kommer också att behandla fördelar och nackdelar med de båda systemen.

### **1.3 Målgrupp**

Målgruppen för det här arbetet är väldigt bred, alla som är intresserade av webbutveckling med hjälp av CMS-system. Arbetet kommer att gå grundligt in på hur man bygger upp en webbplats men också ha mera avancerade detaljer. I designen kommer vi att gå in på responsiv webbdesign vilket är ett väldigt populärt ämne i dagens läge. Vi kommer också att jobba med PHP, MySQL och CSS. Allt det här leder till att arbetet kan vara nyttigt för en nybörjare som är intresserad av webbutveckling t.ex. en förstaårs studerande.

## **2 Val av verktyg**

De flesta verktyg vi använder oss av är sådana som man alltid använder i webbutveckling. Som programmeringsspråk använder vi HTML och PHP samt CSS för designen. Koden skriver vi i programmet Notepad++ och vi använder oss av en MySQL databas. Som stöd i kodandet och i debug-syfte har vi använt oss av browser-modulen FireBug. För

bildmanipulering har vi använt Photoshop. Vi kommer att gå mera noggrannt in på de här olika verktygen nedan.

## 2.1 HTML och CSS

HTML är en förkortning av HyperText Markup Language. Hypertext är helt enkel text som innehåller aktiva hänvisningar till annan text. Hänvisningarna kallas för hyperlänker och de innehåller osynlig information om vart länkarna leder. Länken instruerar webbläsaren vilket html dokument som visas. HTML fungerar med hjälp av HTML element som består av taggar som t.ex. `<html> </html>`. (Staflin 2001, s31-32)

Tim Berners-Lee skrev 1989 ett memo där han föreslog ett Internet baserat hypertext system. 1990 specificerade han HTML. Första gången HTML var tillgängligt för allmänheten var i och med dokumentet "HTML Tags" som Tim Berners-Lee skrivit. HTML 2.0 kom ut 1995 och all utveckling efter det är baserat på just HTML 2.0. Sedan 1996 har HTML utvecklats av World Wide Web Consortium (W3C) som grundades av Tim Berners-Lee.

En av versionerna som används i dagens läge är HTML 4 som släpptes 1999. HTML5 började man arbeta på 2004 och den första versionen kom ut 2008. HTML5 är ännu inte helt färdigt men det förväntas bli helt klart någon gång under 2014. Det går helt bra att använda HTML5 och det används mycket till exempel för att visa videon på mobiltelefoner och pekplattor.

CSS står för Cascading Style Sheets som kan översättas ungefär till överlappande stillistor men som på svenska brukar kallas bara stillistor eller mera allmänt det mycket enkla och internationella CSS. (Korpela 2003)

CSS används för att beskriva hur ett strukturerat dokument, såsom HTML eller XHTML, ser ut. Alltså layouten, färgerna, textstorleken och typsnitten som skall användas på en webbplats. Den viktigaste uppgiften för CSS är att dela upp dokumentinnehållet som är skrivet i HTML från hur dokumentet skall se ut. (Korpela 2003 k. 1)

CSS kan används också för att bestämma hur en sida skall se ut på olika format som t.ex. olika webbläsare, på en mobiltelefon, på printat papper eller på datorer som använder olika

resolution. Detta gör man så att webbsidan skall se så bra ut som möjligt i alla möjliga medium. (Korpela 2003, k. 1)

CSS har med tiden blivit allt populärare och anses nuförtiden vara det viktigaste verktyget för webbdesign. (Korpela 2003, k. 1)

Stillistor har funnits ända sedan 1970-talet men man kan säga att CSS uppstod 1994. Internet blev allt populärare men det fanns inget sätt att enkelt designa webbplatserna. Håkon Wium Lie som hade jobbat med program som man använde för att göra design för tidningar insåg att det skulle behövas ett enkelt sätt att göra designen för webben. 1994 publicerade Håkon Cascading HTML Style Sheets eller CHSS, den första versionen av hans stillista. Samtidigt jobbade Bert Bos på en ny webbläsare som han kallade Argo. Det var en mycket anpassningsbar webbläsare som stödde stillistor. Bert Bos och Håkon Wium Lie bestämde sig för att jobba tillsammans och skapade så alltså den första versionen av CSS som publicerades i november 1994.

Den första officiella versionen av CSS, CSS1 publicerades i december 1996. I februari 1997 skapades en egen grupp för CSS i W3C som jobbade på att utveckla CSS vidare. Således publicerades CSS2 i maj 1998. Man startade jobbet på CSS3 1998, men man arbetar ännu med det när detta skrivs.

## **2.2 PHP**

Utvecklingsarbetet av PHP började år 1994 av Rasmus Lerdorf. PHP står för Hypertext Preprocessor men på grund av att den första versionen kallades "Personal Homepage Tool" så blev förkortningen PHP. I juni 1998 slog PHP igenom och blev mycket populärt då den nya versionen PHP 3.0 släpptes.

PHP är först och främst ett verktyg för att skapa dynamiska webbplatser. PHP:s syntax har till huvudsak lånats från programmeringsspråket C++ vilket betyder att det är enkelt att lära sig om man har tidigare erfarenhet av det. PHP är gratis att använda och dess källkod är öppen för alla. PHP används främst med Linux/Unix apache servrar men det går att använda även med Windows t.ex. med en Access databas.

PHP är ett relativt lätt och användarvänligt programmeringsspråk. Eftersom språket är så populärt så stöder alla webbhotell dess användning och det finns också mycket dokumentation och hjälp att få. Man kan hitta lösningar till de flesta problemen på olika forum på nätet. (Rantala 2005, s09-10)

PHP 4.0 släpptes den 22.5.2000 och PHP 5.0 släpptes den 13.7.2004. I dagens läge används huvudsakligen PHP 5 och den nyaste versionen är PHP 5.5 som släpptes den 20.6.2013.

I det här arbetet använder vi oss av PHP främst genom att insticksmodulerna är gjorda med programmeringsspråket PHP. Vi har också gjort en del egna PHP funktioner då det behövs.

### **2.3 MySQL**

Relationsdatabasservern MySQL skapades i maj 1995 från ett internt företagsprojekt gjort av TcX DataKonsult AB från Sverige. Deras projekt släpptes för allmänheten i slutet av 1996. Programvaran blev så framgångsrik att de skapade ett företag baserat helt och hållet runt MySQL 2001. Företaget fick namnet MySQL AB. MySQL AB var fruktansvärt framgångsrikt och de grundade många avdelningar i flera länder. Till slut köptes det av Sun Microsystem 2008 som sedan köptes av Oracle Corporation i början på 2009. MySQL är populärt till stor del på grund av att det är gratis för många användningsändamål men ändå har samma funktioner som alla andra databashanterare. (Gilmore 2010 s. 477)

MySQL är en DBMS som står för DataBase Management System som är väldigt populär i webbutveckling. Det brukar på svenska kallas databashanterare. MySQL används för att hantera data i en databas så som lagring, sökning och uppdatering.

### **2.4 Notepad++ och Firebug**

Notepad++ är en text och källkodseditor som är gjort för Windows. Vi valde att använda det här programmet i stället för Adobe Dreamweaver. I skolan har vi bekantat oss med Dreamweaver men skillnaden mellan de här programmen är inte så stor, så vi valde Notepad++ på grund av att det är gratis och tar upp mindre resurser av systemet.

Firebug är ett webbutvecklingsprogram till webbläsare som gör att man kan kolla och felsöka webbplatsers källkod så som CSS, HTML och JavaScript. Man kan skriva och editera kod i programmet för att testa hur det fungerar på webbplatsen utan att behöva ladda upp till servern. Firebug är ett gratis program med öppen källkod. Firebug är ett mycket populärt program med över tre miljoner aktiva användare dagligen.

### **3 Planering och indelning av arbete**

Vi började arbetet med att komma överens om en gemensam design till webbplatsen vi skall skapa. Vi valde en ganska enkel design eftersom vårt fokus inte var på den biten men vi bestämde oss ändå för att webbplatsen skulle vara responsiv. Efter valet av design bestämde vi oss för vilka funktioner som skulle finnas med på webbplatsen. Detta val gjorde vi på basen av vad som oftast förekommer på nya webbplatser i dagens läge.

Vi började den praktiska delen med att indela arbetet så att Tomas Söderlund gör arbetet i WordPress och Johan Österberg gör arbetet i Drupal. Indelningen skedde mest på erfarenhetsbasis men eftersom båda har tidigare kunskap om båda CMS-systemen kunde vi dela upp arbetet på det här viset och ändå förstå hur webbplatsen har utvecklats. Vi var från första början överens om denna uppdelning eftersom det skulle ha blivit mycket överlops jobb att arbeta tillsammans på båda webbplatserna.

### **4 CMS-system**

CMS är en förkortning av det engelska Content Management System, vilket översatt till svenska blir innehållshanteringssystem. Det används för att hantera och publicera olika former av elektronisk media. CMS används ofta synonymt med WCMS som står för Web Content Management System, på svenska webbpubliceringssystem. De flesta WCMS system är baserade på öppen källkod vilket betyder att man kan utveckla dem som man behöver. WCMS är det som vi kommer att använda oss av i det här arbetet med systemen Drupal och WordPress.

Webbpubliceringssystem används huvudsakligen för att göra utvecklingen av webbplatser förenklad. Förut skrev man webbplatser helt manuellt i HTML och laddade upp dem till en

webbserver. Efter att WCMS system kom med i bilden så blev det möjligt att skapa webbplatser utan att egentligen skriva källkod över huvud taget. Detta gäller bara enkla webbplatser, på mera komplicerade behövs det alltid egen källkod av någon sort. Ett WCMS system förenklar ändå alltid arbetsprocessen oberoende av hur stor webbplatsen är eftersom man kan undvika extra arbete med hjälp av insticksmoduler.

#### 4.1 Insticksmoduler

Den huvudsakliga fördelen med att använda ett webbpubliceringssystem är att det finns så många färdiga insticksmoduler. Det betyder att det finns en insticksmodul för nästan alla tänkbara funktioner. En av fördelarna med insticksmoduler är att de har öppen källkod så det går att modifiera dem enligt behov. Vi kommer att förklara olika insticksmodulers funktion mera noggrant i den praktiska delen.

#### 4.2 Drupal

Grunden till Drupal var ett internt nätverk som Dries Buytaert och Hans Snijder byggde när de studerade i University of Antwerp år 2000. Att ha en bra internetuppkoppling var dyrt så de bestämde sig för att dela på en ADSL uppkoppling mellan åtta studerande för att spara pengar. För att kunna diskutera och dela saker med varandra över nätverket skapade Dries en simpel webbplats med ett inbyggt diskussionsforum.

Efter att Dries blev utexaminerad bestämde vännerna sig för att sätta upp webbplatsen på nätet så att de kunde fortsätta hålla kontakt. De tyckte att *dorp* som betyder by på flamländska var ett passande namn för deras lilla gemenskap men när Dries skulle kolla om domainnamnet *dorp.org* var upptaget gjorde han ett stavfel och skrev istället *drop.org*.

Drop.org utvecklades snabbt till ett forum där ny webbt Teknologi diskuterades och prövades. I januari 2001 släppte Dries källkoden bakom *drop.org* så att alla skulle kunna utnyttja den som en plattform för webbutveckling. I samband med det här ändrades namnet till Drupal som kommer ifrån det engelska uttalet av det flamländska ordet *druppel*, vilket betyder droppe.

Drupal utvecklades snabbt till ett av de mest populära CMS systemen och idag är Drupal det tredje mest använda CMS systemet efter WordPress och Joomla. När detta skrivs är

1,9% av alla webbplatser byggda med Drupal och 5,3% av de webbplatser som är byggda med webbpubliceringssystem använder Drupal. Några exempel på webbplatser som använder Drupal är Twitter och NASA (Drupal (2014)).

I detta arbete använder vi oss av den nyaste versionen av Drupal, Drupal 7 som släpptes den 5 januari 2011.

### 4.2.1 The Drupal Way

Drupal fungerar enligt open-source principen och de är väldigt måna om att de är en gemenskap med starka värden och principer. En av de bästa sidorna med Drupal är att man kan skapa mycket professionella webbplatser utan någon kunskap i att programmera. Detta görs genom att använda moduler som är extremt populära och mycket flexibla. Om man dessutom kan programmera så behöver man bara göra små tilläggsmoduler eller modifikationer på redan existerande moduler istället för att skapa allt från grunden.

Det här tillvägagångssättet att använda Drupal brukar kallas för *the Drupal Way*. Det innebär i praktiken:

- Att man använder stora och välanvända moduler för att lösa så många uppgifter som möjligt.
- Att man använder moduler som utökar de stora och välanvända modulerna istället för att använda moduler som bara uppfyller en uppgift.
- Att problemlösning är fokuserad på hur man kan använda de stora och välanvända modulerna också när man är tvungen att programmera egna funktioner.
- Att man så mycket som möjligt delar med sig av sina lösningar till gemenskapen så att alla andra kan ha nytta av dem.

Tanken bakom det här är att ekosystemet blir bättre av att alla samarbetar och delar med sig än om alla gör sina egna lösningar för sig. Nyttan bakom att bygga webbplatser på det här sättet är dels att det behövs mindre arbete och blir enklare men också att det blir billigare och säkrare. Webbplatser som använder stora och välanvända moduler kommer att kunna utnyttja hela gemenskapens insatser i att hitta buggar och lösningar på problem istället för att vara tvungna att göra allt själv. De stora modulerna utvecklas hela tiden och om det kommer fram något säkerhetsproblem blir de snabbt uppdaterade.



En stor fördel av att använda sig av *the Drupal Way* är också att alla webbutvecklare som känner till Drupal snabbt förstår sig på hur webbplatsen fungerar istället för att det är bara de som själv kodat den som gör det utan att ta tid att sätta sig in i webbplatsens grundfunktioner. (Falk 2011, s. 11-18)

Jag har använt mig av *the Drupal Way* genomgående i utvecklingen av webbplatsen i det här arbetet och jag kommer att ta upp flera av de stora modulerna som utgör grunden i Drupal.

### **4.3 WordPress**

WordPress är ett av det kändaste cms-systemen som används i dagens läge. Populariteten till systemet kan förklaras med att det är lätt att använda för t.ex. bloggar, men också bra att använda till mera konstruerade hemsidor. Cms-systemet är skrivet i php och man använder MySQL som databas. Programmet är kostnadsfritt under licensen GNU. WordPress är byggt med öppen källkod (open source) vilket gör att vem som helst har tillgång till att editera dataprogramkoden. Det finns 100-tals olika insticksmoduler (plugins) och teman till WordPress, men det går också att skräddarsy webbplatser.

WordPress kom ut 27 maj år 2003 av grundarna Matt Mullenweg och Mike Little med bara några rader kod och användar antalet var mycket litet. WordPress är en efterföljare av blogg verktyget b2/cafelog som var gjort av Muchel Valdrighi år 2001. Muchel Valdrighi slutade utveckla b2/cafelog år 2002 men eftersom Matt Mullenweg hade använt hans bloggtjänst ville han skapa en ny tjänst utgående från b2/cafelog vilket fick namnet WordPress.

Det fanns inga insticksmoduler när WordPress kom ut, men år 2004 kom det till moduler med versionen 1.2. WordPress utvidgades snabbt och i dagens läge har WordPress tjänsten laddats ner över 20 miljoner gånger. Den nyaste version idag är WordPress 3.8.1 vilket kommer att användas i det här projektets arbete.

#### **4.3.1 WordPress.com**

Matt Mullenweg som är grundare till företaget Automattic och tjänsten WordPress.org har också skapat en bloggtjänst som heter WordPress.com. Den här tjänsten öppnades 21

november 2005 för allmänheten. Det finns över 60 miljoner WordPress.com sidor i dagens läge, några kända nyhetsbolag som använder dem är CNN och BBC. Den här sidan skall man inte blanda med den separata klienten (WordPress.org), för till den här sidan behöver man inte en egen server utan man använder sig av WordPress.com:s egna. Den här tjänsten är gratis men det finns restriktioner med design, och moduler som kostar.

### **4.3.2 Automattic Ink**

Automattic Ink är ett webbutvecklingsföretag grundat i augusti år 2005 av Matt Mullenweg. Deras största projekt är WordPress.com och WordPress.org. Automattic Ink är ett företag som har 231 arbetstagare runt om i världen. WordPress.com har 131 miljoner unika besökare i månaden vilket är väldigt mycket om man jämför med hur lite arbetstagare det finns på Automattic Ink. Man kan jämföra det här med Yahoo.com:s 103 miljoner unika besökare men 11,700 arbetstagare.

Ett projekt som också är värt att nämnas är insticksmodulen Akismet som är en spam filter modul inbyggd i alla WordPress sidor som har nyare version än 2.0, samt i alla WordPress.com bloggar. Den används också i många andra cms-system, men för att den skall fungera med icke WordPress sidor måste man få en API key till modulen för att den skall kunna kommunicera med Akismet servern.

## **5 Design**

Som nämndes tidigare så var designen inte det viktigaste i det här arbetet. Målet med designen var att få ett stilrent och enkelt utseende på webbplatsen. Skillnaderna mellan utseendet på webbplatserna beror mestadels på de olika insticksmodulernas funktioner. Vi använde ett tomt tema vilket betyder att designen är gjord helt och hållet med egen kod. Det skulle ha gått att göra helt identiska webbplatser men eftersom målet med arbetet är att testa likheter och olikheter i Drupal och WordPress, så använde vi oss av färdiga moduler som gör att sidan inte blir helt identisk.

Fast vi inte hade planerat mycket tid till designen så tog det längre än väntat eftersom vi bestämt oss för att webbplatsen skulle vara responsiv. Det var jobbigare än väntat att göra två funktionellt och visuellt identiska webbplatser när vi producerade varsin sida.

## 5.1 Responsive Web Design

Med Responsive Web Design (RWB) menas att man bygger en webbplats så att den är funktionsduglig på alla tänkbara mobila apparater och datorer. Det fungerar så att webbplatsen dynamiskt krymper med skärmupplösningen så att webbplatsen ser bra ut och är funktionsduglig på alla plattformar.

Termen Responsive Web Design kom till i maj 2010 i en artikel skriven av webbutvecklaren Ethan Marcotte. 2011 skrev han en bok som hette Responsive Web Design och i vilken han förklarade teorin och praktiken bakom ämnet. Före man började använda responsiv design måste man bygga en skild webbplats för mobila plattformar vilket var tidskrävande och därför kostade mera. Eftersom mobila plattformar är så populära nuförtiden så är Responsiv Design ett av de hetaste ämnen inom webbutveckling för tillfället (Marcotte 2014).

Det är väldigt svårt och ibland omöjligt att konvertera en existerande webbplats till Responsive Web Design. Det här betyder att det är viktigt redan i planeringsfasen att bestämma om webbplatsen skall vara responsiv eller inte.

### 5.1.1 Responsiv design i praktiken

Det viktigaste när man börjar utveckla en webbplats är att man använder relativa värden som procent i stället för absoluta värden som pixlar när man definierar storleken på de olika elementen. Det här betyder i praktiken att webbplatsen minskar när skärmupplösningen minskar.

Kodexempel 1 på nästa sida visar hur CSS koden ser ut för responsiv design. Det här visar webbplatsens utseende när den är över 996 pixlar stor, alltså vanlig storlek när man surfar till webbplatsen på en dator. Som man ser är webbplatsen uppbyggd med två regioner, content och sidebar, som är definierade i procent så att de minskar dynamiskt när skärmupplösningen minskar. Figur 1 visar hur det ser ut på webbplatsen.

Kodexempel 1. Kodexemplet visar hur responsiv design fungerar.

```
#wrapper {
  width: 996px;
  margin: 0 auto;
}

#content {
  width: 66.66667%;
  float:left;
}

#sidebar {
  float:right;
  width:33.33333%;
}
```



Figur 1. Bilden visar content och sidebar i fullskärm.

Eftersom webbplatsen inte skulle se bra ut på en liten skärm, som en mobiltelefon, om den bara är krympt så måste man ändra på layouten genom att använda CSS regeln: @media. Den gör att koden som kommer efter tas i bruk först när @media regeln gäller.

Kodexempel 2 visar hur @media regeln ser ut i koden. Den här CSS-koden tas ibruk först när skärmbredden är max 650 pixlar bred. Som synes har alla regioner nu bredd 100% vilket leder till att de kommer att visas under varandra i den mobila versionen av webbplatsen.

#### Kodexempel 2. Kodexemplet visar hur @media regeln fungerar.

```
@media screen and (max-width: 650px) {  
  #wrapper {  
    width: 100%;  
  }  
  #sidebar {  
    width: 100%;  
    margin-top: 5px;  
  }  
  #content {  
    width: auto;  
    float: none;  
    margin: 5px 0;  
  }  
  #headerbalk {  
    display: none;  
  }  
  #mobinavi {  
    display: block;  
    width: 100%;  
    background: #c76631;  
  }  
}
```

Eftersom menyn inte skulle rymmas i mobilversionen så måste man ha en annorlunda meny i den. #headerbalk hänvisar här till den originala menyn och den är nu satt så att den inte syns med *display: none*. Menyn som kommer att synas i mobilversionen är #mobinavi och därför är den här *display: block*. Vi förklarar mera om hur de olika menyerna fungerar när vi går in på de olika insticksmodulerna som vi använt.

Figur 2 på nästa sida visar hur webbplatsens olika regioner ser ut i mobilversionen. Här syns hur sidebar kommer under content istället för bredvid som i fullskärmsversionen. Det syns också hur navigationsmenyn har ändrat.



Figur 2. Bilden visar webbplatsen som den ser ut i mobilformat.

## 6 Installation

Det första steget i att använda ett CMS-system är självklart att installera det. Eftersom installationen är ganska identisk i de båda systemen är det onödigt att förklara två gånger hur det är gjort. Därför bestämde vi oss för att visa noggrant hur installationen fungerar i Drupal. Processen är den samma för WordPress.

Eftersom Drupal är skrivet i PHP och lagrar data i en databas så krävs det en databas för att installera Drupal. Drupal 7 fungerar med MySQL 5.0.15 uppåt, PostgreSQL 8.3 uppåt och SQLite 3.3.7 uppåt. Drupal kräver minst 15 MB men för en större webbplats med många moduler behövs åtminstone 60 MB utrymme. I detta räknas inte med alla bilder och dylikt som tar mycket utrymme utan bara det som Drupal kräver för att fungera. För att installera Drupal behöver man också en webbserver. Drupal fungerar på Apache, Nginx och Microsoft IIS varav Apache är rekommenderat.

Det första steget i installationen är att ladda ner den senaste versionen av Drupal. Den hittar man på drupal.org under knappen Get Started with Drupal. Efter att man packat upp filen är det dags att ladda upp filerna till serverns webbkatalog. Det har ingen betydelse var på servern man placerar filerna. Det går lika bra att sätta dem i rotkatalogen som någon annanstans. Det enda det påverkar är sökvägen till webbplatsen.

För att kunna installera Drupal måste man skapa en databas. Lättast går det här genom att använda ett program som phpMyAdmin. Det säkraste sättet att skapa en databas i phpMyAdmin är att skapa en användare som har alla rättigheter till den nya databasen men inte till andra databaser. Logga in till phpMyAdmin och gå till fliken *Users*. Tryck på *Add user* och fyll i *User name*: som önskat. I de flesta fall lönar det sig att sätta *Host*: som *local*. Välj lösenord och kryssa sedan för *Create database with same name and grant all privileges*. Tryck sedan *Go* så skapas det en användare och en databas. Figur 3 visar hur det ser ut när man skapar en databas.

## Add user

The screenshot shows the 'Add user' form in phpMyAdmin. It is divided into two main sections: 'Login Information' and 'Database for user'.

**Login Information:**

- User name:** A dropdown menu labeled 'Use text field:' is set to 'drupadb'.
- Host:** A dropdown menu labeled 'Local' is selected, and the text field contains 'localhost'.
- Password:** A dropdown menu labeled 'Use text field:' is selected, and the text field contains masked characters (dots).
- Re-type:** A text field contains masked characters (dots).
- Generate password:** A button labeled 'Generate' is next to an empty text field.

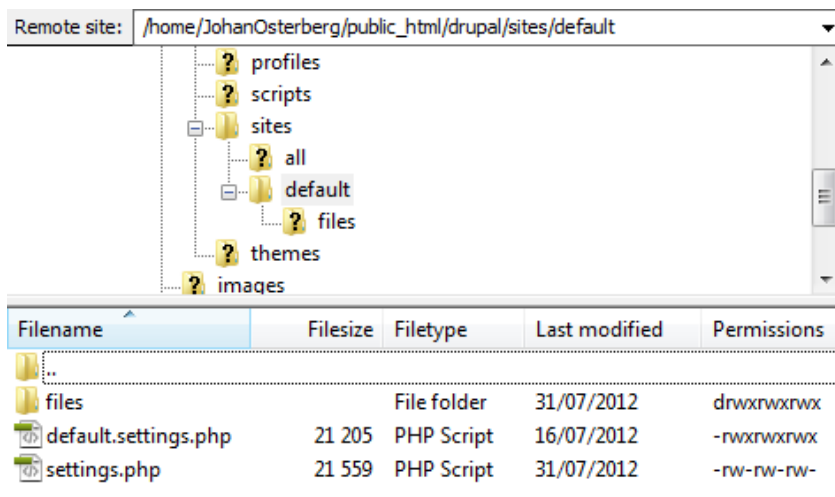
**Database for user:**

- Create database with same name and grant all privileges.
- Grant all privileges on wildcard name (username\\_%).

Figur 3. Exempel på hur man skapar databas.

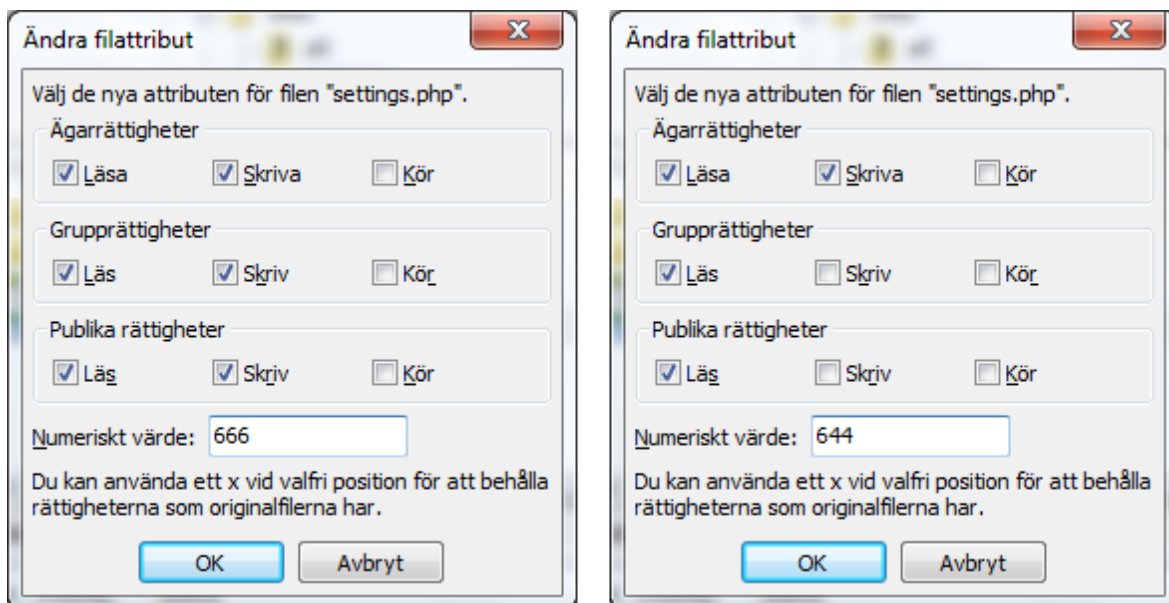
Före man börjar den egentliga installationen måste man göra några förändringar i filerna så att Drupal har de rättigheterna som behövs. I katalogen *sites/default/* finns filen *default.settings.php*, den skall man kopiera och byta namn på till *settings.php*. Det skall alltså finnas en fil som heter *default.settings.php* och en med namnet *settings.php* i katalogen *sites/default*. Det är också en god ide att skapa en katalog med namnet *files* i *sites/default*.

Figur 4 visar hur filerna skall se ut i Filezilla när Drupal är installerat.



Figur 4. Filezilla när man installerar Drupal.

Katalogen *files* och filen *settings.php* måste vara skrivbara för Drupal, vilket betyder att de måste vara skrivbara för webbservern. I ftp-program som Filezilla kan man högerklicka på kataloger och filer och välja *filrättigheter*. I figur 5 går det att se hur filattributen ser ut i Filezilla.



Figur 5. Filattribut i Filezilla.

Bilden ovan till vänster visar hur filrättigheterna är inställda så att det är tillåtet att läsa och skriva till filerna. Man kan antingen kryssa i som bilden visar eller skriva 666 som numeriskt värde. Det är viktigt att komma ihåg att ändra tillbaka de här värdena efter att

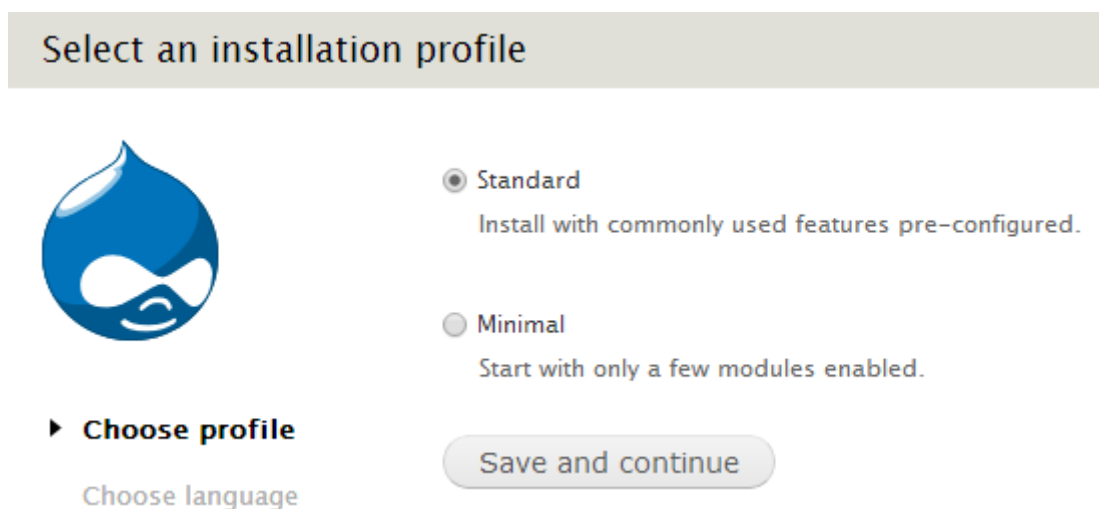


Drupal är färdigt installerat. Bilden ovan till höger visar hur inställningarna skall vara efter installationen.

Nästa steg i installationen är att köra installationsscriptet. Det gör man genom att gå med en webbläsare till den katalog man placerat Drupal i. Installationen borde starta automatiskt men om den inte gör det skriv /install.php efter rotkatalogen.

I det här skedet kommer Drupal att leda användaren igenom flera installationsskeden som beskrivs nedan:


1. Välj profil. Här väljer man mellan standard och minimal profiler. Standardprofilen är den som de flesta borde välja eftersom den innehåller alla de vanliga modulerna som behövs. Minimal är en profil som innehåller bara det som absolut behövs för att köra Drupal. Den rekommenderas bara för erfarna Drupalanvändare som vet exakt vad webbplatsen de bygger kommer att behöva.



Figur 6. Välj profil.

2. Välj språk. Här väljer man vilket språk man vill installera Drupal i. Som standard finns bara engelska och det går bra att sätta till nya språk efter att installationen är gjord.

## Choose language



English (built-in)

[Learn how to install Drupal in other languages](#)


✓ Choose profile

▶ **Choose language**

Figur 7. Välj språk.

3. Efter språkvalet kommer Drupal att kolla att filsystemet är inställt korrekt. Om det inte är det så förklarar installationen vad som är fel så att det kan korrigeras. Om man följt instruktionerna som nämnts tidigare så kommer allt att fungera.
4. Konfiguration av databas. Ange namnet på databasen som du skapade tidigare och ange också användarnamnet och lösenordet för databasen.

## Database configuration



✓ Choose profile

✓ Choose language

✓ Verify requirements

▶ **Set up database**

Install profile

Configure site

Finished

**Database type \***

MySQL, MariaDB, or equivalent

SQLite

The type of database your Drupal data will be stored in.

**Database name \***

The name of the database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

**Database username \***

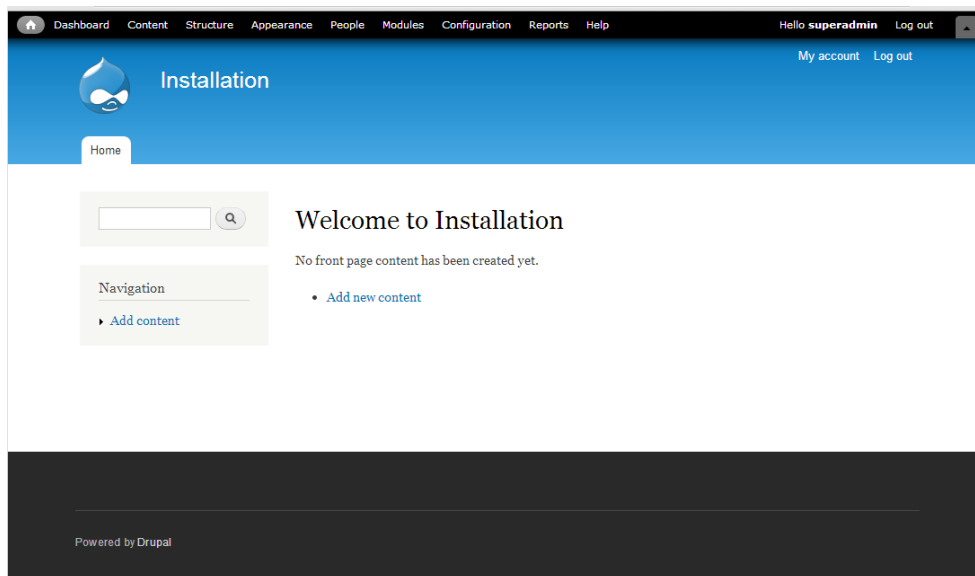
**Database password**

Figur 8. Konfiguration av databas.

5. Efter att databasen är konfigurerad installeras Drupal vilket kan ta några minuter.

6. När installationen är klar skall man ange en del grundläggande information för webbplatsen som webbplatsens namn och en rotanvändare. Här kan man fylla i helt enligt önskemål.

Nu är Drupal färdigt installerat och det är dags att börja bygga webbplatsen.



Figur 9. Drupal installerat.

## 7 Tema

Teman gör det möjligt att ändra hur webbplatsen ser ut. Det finns färdigt gjorda teman till både Drupal och WordPress som det går att använda. Man kan också göra helt egna teman från grunden eller utgående från ett befintligt tema.

### 7.1 Tema i Drupal

För Drupal bestämde jag mig för att använda temat Zen som är det mest populära temat i Drupal om man skall bygga upp en design från grunden. Den enklaste metoden att installera teman i Drupal är att gå till Appearance och trycka på Install new theme. Där kan man klistra in en länk till temats paket som man hittar på Drupal.org i rutan Install from a URL. Drupal laddar då själv ner temat och installerar det på webbservern förutsatt att webbserverns rättigheter är rätt inställda.

För många teman rekommenderas det att man skapar en så kallad sub-theme för att inte ställa till med något i temats rotfiler och för att det går lättare att uppdatera temat så. I Zen-temat gör man det här genom att kopiera katalogen STARTERKIT som finns inne i Zen katalogen ut i "sites/all/themes" och byta namn på den till vad man vill. Namnet får bara innehålla små tecken och siffror. I den nya katalogen finns det en fil med namnet STARTERINFO.info.txt. På den filen skall man ändra namnet så att det är samma som på katalogen den är i och ta bort .txt. Inne i filen skall man ändra "name = Zen Sub-theme Starter Kit" till "name = Ditt valda namn" och "description = Read..." till "description = A Zen sub-theme". Till sist går man till appearance och flushar cachen för teman.

Zen finns i två olika versioner, varav den ena är responsiv. Den versionen är ny sedan jag senast använde Zen och eftersom en av målen med arbetet var att göra webbplatsen responsiv bestämde jag mig för att använda Zens färdiga responsiva layout. Layouten fungerade relativt bra som den var men jag var tvungen att ändra alla procenter och @media storlekar så att de motsvarade vår design.

## 7.2 Tema i WordPress

WordPress kan byggas upp från grunden själv, eller så finns det olika teman att välja mellan. Det finns mera krävande teman som kostar, men också gratis teman. De är uppbyggda på en färdig design och det finns också färdiga insticksmoduler att använda. Det är svårt att ändra designen på ett färdigt tema samt att installera nya moduler vilket kan vara en stor nackdel ifall man vill ändra någonting på webbplatsen.

Till WordPress sidan användes ett så kallat Skeleton tema. Det här temat är ett färdigt tema med olika templates men det är ingen färdig kod insatt i dem. Det betyder att man får designa sidan helt själv från början. Fördelen med det här är att man får webbplatsen helt uppbyggt på eget sätt och man kan göra lätta ändringar på webbplatsen om det så behövs. Temat har heller inte några insticksmoduler färdigt inbyggda, så de får man installera manuellt själv.

För att installera ett tema i WordPress laddar man ner det, laddar upp det i temakatalogen på webbservern och aktiverar det under Themes i Appearance.

## 8 Installera moduler i Drupal

Det enklaste sättet att installera nya moduler i Drupal är att:

1. Gå till [Drupal.org](http://Drupal.org) och söka upp den modul man vill installera.
2. Kopiera länken till modulens arkivfil som är i formatet *tar.gz*.
3. Gå till Install new modules under modules på din webbplats och klistra in länken i Install from a URL fältet.
4. Tryck Install så installeras modulen automatiskt till webbservern.
5. Gå till modullistan och aktivera den nya modulen genom att kryssa i den i listan och tryck på Save Configuration. Modulen är nu installerad.

Det här systemet fungerar ifall modulen du vill installera inte har några andra krav. Om du inte efter steg 5 hittar modulen eller det inte går att aktivera den så betyder det att modulen antingen inte har något gränssnitt eller att den är beroende av någon annan modul. Det bästa sättet att ta reda på vad som behövs är att granska vad det står i modulens readme om förhandskrav och följa dem.

### 8.1 Installera moduler i WordPress

Alla Wordpressmoduler finns på webbplatsen [WordPress.org](http://WordPress.org). De finns under rubriken Plugins. När man öppnar webbplatsen kommer det en lista upp på alla populära instickmoduler så att man själv kan söka på vad man letar efter. Varje modul har instruktioner till hur man installerar den, men installationsprocessen går i grund och botten till på följande sätt.

1. Ladda ned insticksmodulen som kommer i en zip fil.
2. Packa upp filen till din dator.
3. Överför sedan filen från datorn till servern till katalogen plugins.
4. Gå in till din Wordpress sida och aktivera din instickmodul vid Plugins.

När du har följt de här stegen kan du börja redigera inställningarna på din modul och om det är en widget modul så överför du den till det widget fält du vill att den skall synas i.

## 9 Block/Widget

För att få funktioner att synas i en container använder man sig av någonting som kallas för blocks i Drupal och widgets i WordPress. Många insticksmoduler visas som blocks och widgets.

### 9.1 Block i Drupal

Block är en modul som ingår i Drupals grundinstallation. Det betyder att den finns färdigt tillgänglig när man börjar arbeta på en ny webbplats. Att använda sig av block är alltså mindre komplicerat än widget i WordPress.

Att placera ut blocken dit man vill är enkelt. Det är bara att navigera till Block under Structure. Där hittar man en lista på alla regioner som är definierade i temat och en lista på alla block som är installerade. Listan fungerar med *drag and drop* metoden, d.v.s. man drar in blocken till den region man vill placera dem i. För att få blockena att passa in snyggt i webbplatsens design kan man använda sig av CSS attribut som automatiskt tilldelas alla block i Drupal.

### 9.2 Widget i WordPress

Med hjälp av widgets kan man tillägga mycket olika funktioner till din sidebar, footer eller header. Widgets var från början gjort för att få en lätt och bra struktur över alla funktioner till WordPress. För att använda widgets så behöver man inte någon erfarenhet av kodning, utan man kan aktivera eller deaktivera widgets i Appearance ⇔ widget panelen.

När man bygger en sida från grunden i WordPress så måste man aktivera widgets och sidebaren. Det gör man i filen Function.php med en kod som följer på nästa sida.

### Kodexempel 3. Aktivering av Widgets och sidfältet.

```
function arphabet_widgets_init() {

    register_sidebar( array(
        'name' => 'sidebar',
        'id' => 'home_right_1',
        'before_widget' => '<div>',
        'after_widget' => '</div>',
        'before_title' => '<h2 class="rounded">',
        'after_title' => '</h2>',
    ) );

}

add_action( 'widgets_init', 'arphabet_widgets_init' );
```

I kodexempel 3 syns det hur koden registrerar din sidebar vilket på samma gång aktiverar din widget panel till WordPress sidan. Om du sedan vill aktivera flera sidebars till widgets panelen så är det bara att kopiera Register\_sidebar koden med ett ändrat "name" och "id". För att få sidfältet att synas på din webbplats måste du också tillägga en rad kod i din template php.

### Kodexempel 4. Aktiveringskoden i templatefilen.

```
<?php dynamic_sidebar( 'sidebar' ); ?>
```

Den här koden ska du lägga in i vilken container du vill ha dina widgets att synas i. Självt har jag lagt mina widgets att visas i min sidebar container, så min kod ser ut såhär i template filen.

### Kodexempel 5. Slutliga koden.

```
<div id="sidebar">
    <?php dynamic_sidebar( 'sidebar' ); ?>
</div>
```

## 10 Navigationsmeny

Alla webbplatser behöver en navigationsmeny som är tydlig och fungerar bra så att webbplatsens besökare säkert hittar allt de vill. Vi bestämde oss för att navigationsfältet

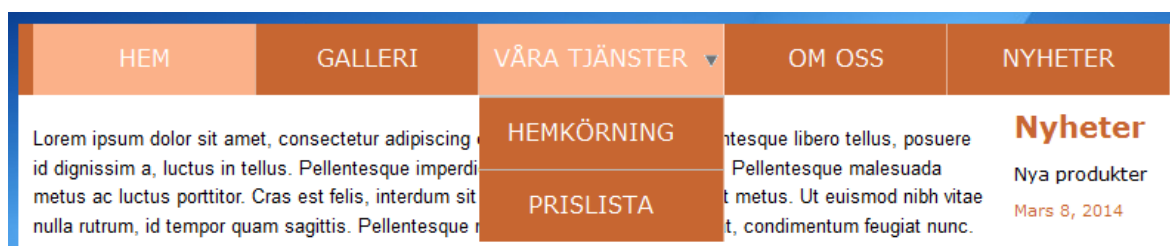
skulle fungera som rullgardinsmeny vilket betyder att underrubrikerna är gömda tills man flyttar muspekaren över huvudrubriken.

Eftersom webbplatsen skall vara responsiv måste menyn ändra format när skärmupplösningen blir för liten. Skillnaden mellan hur vi gjorde den responsiva menyn var att i Drupal användes en modul för det medan i WordPress skapade vi den responsiva menyn själva med en PHP-funktion. Det finns moduler till WordPress som sköter den responsiva navigationen automatiskt men vi ville lära oss hur man gör det från grunden.

## 10.1 Navigation i Drupal

Drupal har ett inbyggt navigationssystem men det är onödigt komplicerat att utgå från det om man vill ha en rullgardinsmeny. Det finns många moduler för menyer att välja mellan men jag valde Nice Menus för att den har de funktioner jag behövde utan att vara alltför stor som en del andra ännu mera avancerade moduler som Mega Menu.

Nice Menus fungerar så att det skapar nya block som man kan placera ut var man vill. I de här blocken kan man välja en av Drupals menyer eller skapa en egen meny som blocket kommer att visa. Jag använde mig av Drupals main menu och satte in alla sidor som skall vara i menyn dit. Blocket placerade jag i regionen navigation som är under header-regionen. Sedan kan man designa menyn med CSS att se ut som man vill.



Figur 10. Nice Menus slutliga utseende.

För att översätta menyn använde jag tilläggsmodulen Menu Translation. Menyn översätts sedan genom att navigera till menyn som skall översättas under Structure/Menus. Där går det att välja translate på varje menylink. Efter att alla menylinkar översatts måste linkarna arrangeras på alla språk, det görs helt normalt med drag/drop metoden i menylistan.



### 10.1.1 Responsiv navigation

Eftersom menyn är för bred när man besöker webbplatsen med en telefon så måste man ändra den då skärmbredden blir mindre. För att uppnå det här har jag använt en modul som heter Responsive Menu. Responsive Menu kan göra vilken meny eller HTML-lista som helst responsiv. I Responsiv Menu inställningar skriver man in vilka menyer man vill göra responsiva och så sköter modulen resten.

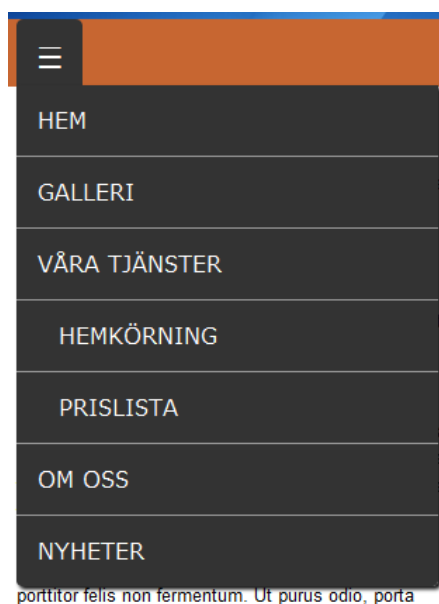
#### Selectors for which menus to responsify

ul.nice-menu

Enter CSS/jQuery selectors of menus to responsify. Comma separated or 1 per line

Figur 11. Lista över vilka menyer som skall göras responsiva.

Responsive Menu har färdigt installerat fyra olika stilar för den responsiva menyn av vilka jag har valt Simple Expanding. Jag valde den för att den passade in i webbplatsens layout efter en del CSS kodning.



Figur 12. Den responsiva menyn.

Figur 12 visar hur den responsiva menyn ser ut i mobilformat.

## 10.2 Navigation i WordPress

Jag har valt att använda Dropdown Menu Widget vilket är en av de kändaste navigationsfälten som WordPress har att erbjuda. Insticksmodulen har över 290 000 nerladdningar vilket säger en hel del. Den här modulen har ett väldigt bra redigeringsfält i WordPress. Jag valde den här insticksmodulen eftersom navigationsfältet skulle vara i rullgardinsformat. Den här modulen har en färdig rullgardinsfunktion vilket gjorde arbetet väldigt lätt.

För att få modulen att visas skall man lägga till en rad php kod i den container man vill att den att synas, jag själv ville att den skulle visas inom min navigationsbalk som heter headerbalk så koden ser ut så här för att visa den där.

**Kodexempel 6. Kod för att placera in navigationsmodulen.**

```
<div id="headerbalk"><?php shailan_dropdown_menu(); ?> </div>
```

Jag behövde inte koda någonting själv för att ändra på modulens funktioner, utan kunde bara använda Dropdown menu widgetens egna redigerings fält för att få allt att fungera. I redigeringsprogrammet kunde man ändra på bl.a. effekterna, färgerna och sedan lite mera avancerade inställningar.

The image shows the 'Custom colors' panel in the WordPress Customizer. At the top, there are tabs for 'General', 'Template Tag', 'Effects', 'Custom colors', and 'Advanced'. Below the tabs, a yellow banner reads: 'Using options below you can customize certain elements of current theme. If you select **Color Scheme** as your theme...'. The panel contains several settings:

- Use custom colors:** A checked checkbox with a note: 'If not checked custom colors won't work. [?]'
- Menu Background Color:** A color picker set to #C76631. Description: 'Background color of the dropdown menu [?]'
- Hover Background Color:** A color picker set to #fbb189. Description: 'Background color of list item link. [?]'
- Link Text Color:** A text input set to #FFFFFF. Description: 'Default link color [?]'
- Link Text Color on mouse over:** A text input set to #FFFFFF. Description: 'Secondary link color [?]'
- Overlay:** A dropdown menu set to 'none'. Description: 'Menu overlay (Works on browsers that support png transparency only.) [?]'

**Figur 13. Redigeringsprogrammets utseende.**

### 10.2.1 Responsiv navigation

För att få navigationen responsiv måste man byta ut den originella navigationsbalken med en responsiv navigering. I CSS filen fungerar det så att man gömmer den originella navigationsbalken när skärmbredden blir under 650px. Det betyder att när @Media blir under 650px så blir originella navigationen Display:none och den responsiva display:block.

I kodexempel 7 ser man att den originella navigeringen är i #headerbalk, därför måste man gömma den så att #mobinavi som är responsiv syns

Kodexempel 7. Kod för att visa den responsiva navigationen.

```
@media screen and (max-width: 650px) {  
  #headerbalk {  
    display: none;  
  }  
  #mobinavi {  
    display:block;  
    width: 100%;  
    background: #C76631;  
  }  
}
```

Det responsiva navigeringsfältet fungerar genom en php kod som är insatt i header template filen innanför #mobinavi regionen. Koden ser ut som följande.

## Kodexempel 8. Kod som skapar en navigeringslista.

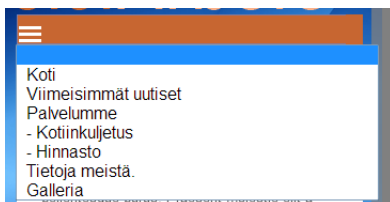
```

<div id="mobinavi">
  <div class="styled-select">
    <select name="page-dropdown"
      onchange='document.location.href=this.options[this.selectedIndex].value;'>
      <option value=""></option>
      <?php
        $args = array (
          'sort_order' => 'ASC',
          'sort_column' => 'menu_order'
        );
        $pages = get_pages($args);
        foreach ( $pages as $page ) {
          $option = '<option value="' . get_page_link( $page->ID ) . "'>';
          if($page->post_parent!="0") {
            $option .= "- ";
          }
          $option .= $page->post_title;
          $option .= '</option>';
          echo $option;
        }
      ?>
    </select>
  </div>
</div>

```

Den här koden gör en ny navigationsbalk med rullgardinseffekt. Härfter förklarar jag vad koden gör.

- Koden har klassen `.style-select` med vilket jag har designat utseende på navigationen i css filen.
- Select name skapar en klickbar lista.
- Onchange gör att när man klickar på ett av alternativen i listan så fungerar det som en länk.
- \$args skapar en array som sorterar listan.
- \$pages = get\_pages(\$args) sätter in alla sidor i arrayn.
- foreach är en loop som tar namnen och länkarna från sidorna och sätter till dem till listan. if-satsen gör så att underrubriker får ett "-" före namnet.



Figur 14. Responsiva navigationen i WordPress.

## 11 Flerspråkig webbplats

Webbplatsen skall vara på både finska och svenska eftersom det är standard i Finland. Det är ganska stor skillnad i hur man implementerar det i Drupal och WordPress. WordPress kräver mindre arbete medan det ännu i Drupal 7 är en komplicerad process att översätta webbplatser.

Den stora skillnaden mellan Drupal och WordPress beror på att man i Drupal behöver många moduler för att översätta allt innehåll på webbplatsen. I WordPress behövs bara en modul för den här webbplatsen.

### 11.1 Flerspråkig webbplats i Drupal

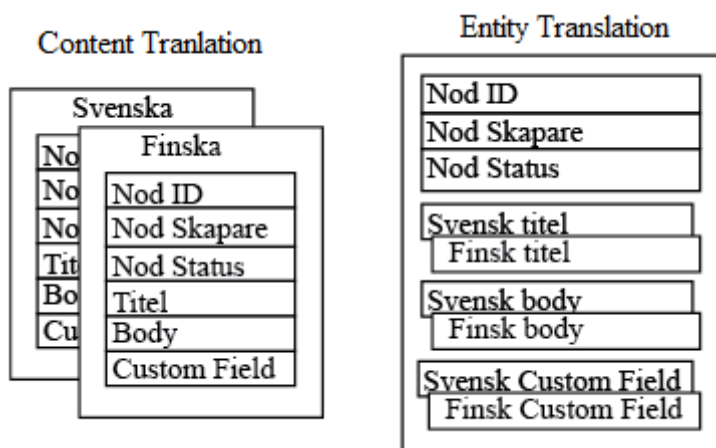
Ett problem Drupal har är att det inte är riktigt enkelt att skapa en webbplats som skall vara flerspråkig. Man måste använda flera olika moduler för att det skall fungera bra istället för att det skulle finnas en modul som sköter allting.

Det finns två olika komponenter som man måste tänka på när man översätter en webbplats med Drupal. De är gränssnittet och innehållet. Gränssnittet är allt det som är inbyggt i Drupal (som datum och dylikt), sådant som är likadant på alla sidor oberoende innehåll. Drupal har en modul inbyggd i standard installationen kallad *Locale* som tar hand om att översatt gränssnittet. Med hjälp av den modulen kan man ladda ner färdiga översättningar av gränssnittet till väldigt många språk. Det varierar väldigt mycket hur stor del av gränssnittet som är översatt. På finska är för tillfället 39% översatt och på svenska 18%. Det går att översätta det som inte är översatt ifall det är någonting som man behöver.

Innehållet kräver mera arbete att översätta. I tidigare versioner av Drupal fungerade översättning av innehållet så att man skapade en egen nod för varje språk som sedan kopplas tillsammans så att Drupal vet vilka noder som är översättningar av varandra. Det här systemet kallas för *Content Translation*.

I Drupal 7 finns ett nytt system för att översätta innehåll. I det här systemet finns det bara en nod för alla språk men alla fält i noden kan ha flera kopior på olika språk. Koden för det här systemet finns i Drupal 7 men det fanns inte tillräckligt mycket tid före Drupal 7 släpptes för att skapa ett gränssnitt för att använda systemet. Därför skapades en modul

som heter *Entity Translation* som gör att det går att utnyttja det nya systemet. Det här betyder att det i Drupal 7 finns två olika metoder för att översätta innehållet på webbplatsen. Om man vill kan man till och med använda båda på samma webbplats. Eftersom det är nyare och jag inte använt det tidigare bestämde jag mig för att använda *Entity Translation*. Figur 15 illustrerar skillnaden mellan *Content* och *Entity Translation*.



Figur 15. Skillnaden mellan Content och Entity Translation.

Som tidigare nämnt måste man använda flera olika moduler för att översätta en webbplats. För det här arbetet använde jag mig av de här modulerna:

- Locale
- Entity Translation
- Title
- Entity
- Internationalization (i18n)
- Views Internationalization
- Menu Translation
- Variable
- Localization Update
- Language Icons

Först aktiverar man på modulerna Locale och Localization Update. Efter det skapas två nya rubriker under administrationen, Language och Detection and Selection. Under Language sätter man till alla språk som webbplatsen skall ha.

- [Add language](#)

[Show row weights](#)

| English name | Native name | Code | Direction     | Enabled                             | Default                          | Admin                 | Operations                                  |
|--------------|-------------|------|---------------|-------------------------------------|----------------------------------|-----------------------|---|
| + English    | English     | en   | Left to right | <input checked="" type="checkbox"/> | <input checked="" type="radio"/> | <input type="radio"/> | <a href="#">edit</a>                        |
| + Finnish    | Suomi       | fi   | Left to right | <input checked="" type="checkbox"/> | <input type="radio"/>            | <input type="radio"/> | <a href="#">edit</a> <a href="#">delete</a> |
| + Swedish    | Svenska     | sv   | Left to right | <input checked="" type="checkbox"/> | <input type="radio"/>            | <input type="radio"/> | <a href="#">edit</a> <a href="#">delete</a> |

Figur 16. Language.

Figur 16 visar hur det ser ut när jag har lagt till Svenska och Finska till Webbplatsen. English är administrationspråk eftersom jag föredrar att ha gränssnittet på engelska när jag arbetar med webbplatsen.

Det är viktigt att gå till Detection and Selection för att ställa in hur webbplatsen vet vilket språk som skall visas. Här finns flera olika alternativ men för de flesta webbplatser är URL det bästa valet. URL betyder att det till adressen läggs till sv för svenska och fi för finska.

| Detection method | Description  | Enabled                             | Operations                |
|------------------|--|-------------------------------------|---------------------------|
| + URL            | Determine the language from the URL (Path prefix or domain). | <input checked="" type="checkbox"/> | <a href="#">Configure</a> |
| + Session        | Determine the language from a request/session parameter.     | <input type="checkbox"/>            | <a href="#">Configure</a> |
| + User           | Follow the user's language preference.                       | <input type="checkbox"/>            |                           |
| + Browser        | Determine the language from the browser's language settings. | <input type="checkbox"/>            |                           |
| + Default        | Use the default site language (English).                     | <input checked="" type="checkbox"/> |                           |

Figur 17. Language Detection and Selection.

Nästa steg är att installera modulerna Entity Translation, Title, Internationalization (i18n), Entity och Variable. Entity och Variable behövs för att en del av de här modulerna behöver dem för att fungera. Title gör så att en nods titel kan konverteras till ett fält så det går att översätta det. Internationalization är ett helt paket med olika moduler för översättning. I det här skedet behövs bara grundmodulen Internationalization och modulen Field Translation.

För att det skall gå att översätta en Content Type skall man gå till Publishing Options under den Content Typen man vill översätta och kryssa i *Enabled, with field translation* under *Multilingual Support*.

|   |  |
|---|--|
| <b>Submission form settings</b><br>Title                            | <b>Default options</b><br><input checked="" type="checkbox"/> Published<br><input type="checkbox"/> Promoted to front page<br><input type="checkbox"/> Sticky at top of lists<br><input type="checkbox"/> Create new revision<br><small>Users with the <i>Administer content</i> permission will be able to override these options.</small>  |
| <b>Publishing options</b><br>Published                              |  |
| <b>Display settings</b><br>Don't display post information, Hidden   |  |
| <b>Comment settings</b><br>Hidden, Threading , 50 comments per page |  |
| <b>Menu settings</b>  |  |
|   | <b>Multilingual support</b><br><input type="radio"/> Disabled<br><input type="radio"/> Enabled<br><input checked="" type="radio"/> Enabled, with field translation<br><small>Enable multilingual support for this content type. If enabled, a language selection field will be added to the editing form, allowing you to select from one of the <a href="#">enabled languages</a>. If disabled, new posts are saved with the default language. Existing content will not be affected by changing this option.</small> |

Figur 18. Flerspråkiga noder.

Sedan skall man gå till Manage Fields för att ställa in vilka fält som skall gå att översätta. För varje fält som skall översättas måste man trycka på edit och gå ner till *Field translation* och kryssa i *Users may translate all occurrences of this field*.

## Field translation

Users may translate all occurrences of this field:

- Text in *Translation*

Figur 19. Field translation.

För att det skall gå att översätta titeln på noden måste man klicka replace under Manage Fields så att titeln blir ett fält.

| Label   | Machine name | Field type          | Widget | Operations              |
|---------|--------------|---------------------|--------|-------------------------|
| + Title | title        | Node module element |        | <a href="#">replace</a> |

Figur 20. Replace titel.

Efter de här stegen kan man översätta noder genom att trycka på Translate på en befintlig nod vilket visar en lista över alla språk som finns på webbplatsen. Här kan man se vilka



som är översatta och vilka som saknar översättning. Webbplatsen kommer automatiskt att visa noden på det språk som användaren väljer.

| Language       | Source language    | Translation          | Status         | Operations           |
|----------------|--------------------|----------------------|----------------|----------------------|
| English        | n/a                | n/a                  | Not translated | <a href="#">add</a>  |
| Finnish        | Swedish            | <a href="#">Koti</a> | Published      | <a href="#">edit</a> |
| <b>Swedish</b> | (original content) | <a href="#">Hem</a>  | Published      | <a href="#">edit</a> |

Figur 21. Lista över översättningar på en nod.

När man skapar en nod måste man välja ett originalspråk för att det skall gå att översätta den.








För att användarna skall kunna välja språk har jag använt modulen Language Icons som skapar ett block med ikoner för de olika språken. Blocket är placerat i sidebaren.

Förutom de här modulerna behövde jag ytterligare två moduler för att kunna översätta allting på webbplatsen, Views Internationalization och Menu Translation, hur dessa moduler fungerar tar jag upp under sektionerna för views och meny.

## 11.2 Språkval i WordPress

Till språket använde jag en insticksmodul som heter qTranslate. Det är den kändaste gratis insticksmodulen som finns. Det finns dock en bättre översättnings insticksmodul som heter The WordPress Multilingual Plugin (WPML) som de flesta företagen använder men den kostar att använda så därför valde jag qTranslate. Till qtranslates grund installation finns det 17 fungerande språk att välja mellan, sedan kan man också tillsätta flera manuellt om det så behövs.

Modulen fungerar så att när man installerat den går man in i inställningarna för språk och väljer vilka språk man vill använda.

| Flag  | Namn       | Aktivitet |      |         |
|---|------------|-----------|------|---------|
|  | Deutsch    | Enable    | Edit | Delete  |
|  | English    | Enable    | Edit | Delete  |
|  | 中文         | Enable    | Edit | Delete  |
|  | suomi      | Disable   | Edit | Delete  |
|  | Français   | Enable    | Edit | Delete  |
|  | Nederlands | Enable    | Edit | Delete  |
|  | Svenska    | Disable   | Edit | Default |

Figur 22. I bilden syns hur finska och svenska är valda.

För att webbplatsen skall veta vilket språk som skall visas måste man ställa in hur webbplatsen bestämmer det. I figur 23 syns att Use Query Mode är ikryssat vilket betyder att det i addressfältet kommer till en textsträng i formatet ?lang=sv om det är på svenska och ?lang=fi om det är på finska.

**Detect Browser Language**  Detect the language of the browser and redirect accordingly.  
When the frontpage is visited via bookmark/external link/type-in, the visitor will be forwarded to the correct URL for the language specified by his browser.

**Advanced Settings**

**URL Modification Mode**

Use Query Mode (?lang=en)

Use Pre-Path Mode (Default, puts /en/ in front of URL)

Use Pre-Domain Mode (uses http://en.yoursite.com)

Figur 23. Inställning för URL val av språk.

För att få språkväljaren att synas på sidfältet måste man gå in i widgets och flytta över språkväljaren till sidfältet. I språkväljarens inställningar kan man välja fyra olika sätt hur språkväljaren ser ut. Jag har valt att visa bara med flaggbilder.

sidebar

Recent Posts Extended

qTranslate språkväljare

Title:

Hide Title:

Display:

Text only

Image only

Text and Image

Dropdown Box

[Delete](#) | [Close](#) [Save](#)

Figur 24. Widgeten för språkval.

qTranslate kunde dock inte översätta widget rubriken till båda språken på sidebaren. För att få språket att fungera på sidebar widget rubriken var jag tvungen att lägga till en kod i PHP. Det är en ganska simpel kod, eftersom det bara finns två språk på sidan. Kodens uppgift är att översätta rubriken till svenska och finska.

**Kodexempel 9. PHP kod för att få widgetrubriken översatt.**

```
<?php $lang=get_bloginfo("language");  
if ($lang=="sv-SE") { ?>  
  <h1>Senaste nyheter</h1>  
  <?php } else { ?>  
  <h1>Viimeisimmät uutiset</h1>  
  <?php } ?>
```

Den här koden betyder att när webbplatsen visas på svenska så syns rubriken Senaste Nyheter, men om det är något annat språk så syns Viimeisimmät uutiset.

## 12 Mera funktionalitet till noderna

Här kommer vi att beskriva en metod i de båda CMS systemen för att få bättre möjlighet att påverka hur noderna visas. När man lagar en sida på webbplatsen är det alltid svårt att få noders layout att se ut som man vill. Så vi ville göra en funktion som gör att när man skriver en page så kan man välja en bild som automatiskt placerar sig på högra sidan av texten.

Modulerna vi använt oss av är ganska olika men båda är lätta att använda. I Drupal behövdes ingen egen kod men i WordPress använde vi en PHP kod för att få bilderna att synas.

### 12.1 Panels i Drupal

Panels är en mycket flexibel insticksmodul med vilken det går att åstadkomma en mängd olika saker. I grund och botten gör Panels det möjligt att påverka layouten på webbplatsen. Det här kan göras på flera olika sätt, Panels kan skapa egna sidor och noder där data tas in enligt önskan. Det jag valt att göra med Panels är att ändra layouten på en content typ.

För att göra det här krävs förutom modulen Panels modulen Page Manager som är en del av paketet Chaos Tools. Efter att dessa två moduler aktiverats skapade jag en ny content typ med namnet Textbild. I den lade jag till ett Image fält där det går att ladda upp två bilder.

För att få bilderna att visas bredvid texten i ett tvåkolumnsformat går man till Pages under Structure i administrationsmenyn. Den här sidan har skapats av modulen Page Manager och den gör det möjligt att skapa sidor och kustomisera dem på olika sätt. Figur 25 visar hur jag aktiverat System node\_view och trycker på edit.

| Type   | Name          | Title                  | Path                          | Storage | Operations               |
|--------|---------------|------------------------|-------------------------------|---------|--------------------------|
| System | comment_reply | Comment Reply page     | /comment/reply/%node          | In code | <a href="#">Enable</a> ▾ |
| System | node_edit     | Node add/edit form     | /node/%node/edit              | In code | <a href="#">Enable</a> ▾ |
| System | node_view     | Node template          | /node/%node                   | In code | <a href="#">Edit</a> ▾   |
| System | search-node   | Content                | /search/node/!keywords        | In code | <a href="#">Enable</a> ▾ |
| System | search-user   | Users                  | /search/user/!keywords        | In code | <a href="#">Enable</a> ▾ |
| System | term_view     | Taxonomy term template | /taxonomy/term/%taxonomy_term | In code | <a href="#">Enable</a> ▾ |
| System | user_edit     | User edit template     | /user/%user/edit              | In code | <a href="#">Enable</a> ▾ |
| System | user_view     | User profile template  | /user/%user                   | In code | <a href="#">Enable</a> ▾ |

- [» Create a new page](#)

Figur 25. Page Manager i Drupal.

I figur 26 klickar jag på Add variant och skriver in en titel. Variant type skall vara Panel eftersom jag vill använda Panels och Selection rules skall kryssas i.

**Add variant**

Add a new variant to this page.

**Title**  
  
Administrative title of this variant. If you leave blank it will be automatically assigned.

**Variant type**

**Optional features**

Selection rules

Contexts

Check any optional features you need to be presented with forms for configuring them. If you do not check them here you will still be able to utilize these features once the new page is created. If you are not sure, leave these unchecked.

Figur 26. Add variant i Page Manager.

Efter att ha klickat på Create variant skall Selection rules väljas. Det ställer in vad som skall visas i den här sidvarianten. Jag väljer Node: type eftersom jag vill ändra layouten på endast en content typ.

| Title                                      | Description |
|--|-------------|
| No criteria selected, this test will pass. |             |
| Node: type                                 | Add         |

All criteria must pass.  
 Only one criteria must pass.

Continue

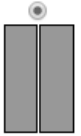

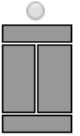
Figur 27. Selection rules i Page Manager.

När man klickar på Add skall man kryssa i vilken eller vilka content typer som skall påverkas. Nästa steg är att välja layout, det går att skapa helt egen layout med så många rader och kolumner man vill ha men jag väljer två kolumner eftersom det är det jag vill ha.

Category

Columns: 2 ▾

Columns: 2

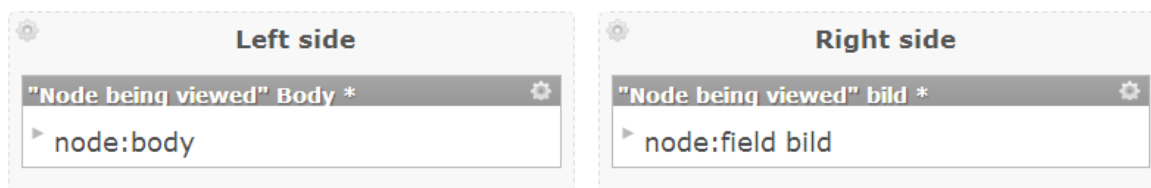
|   |   |   |
|---|---|---|
|  |  |  |
| Two column  | Two column bricks   | Two column stacked  |

Back Continue

Figur 28. Layout i Page Manager.

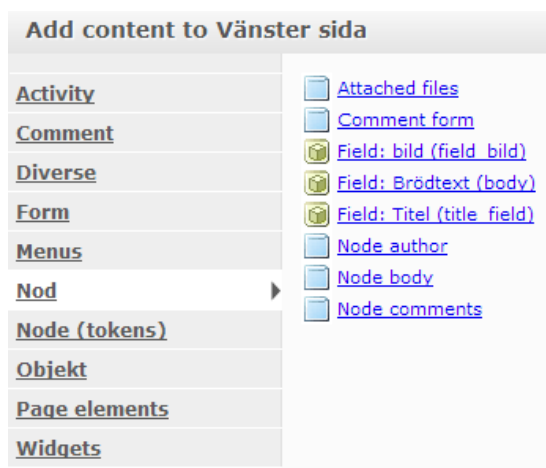
Nu kommer det en sida där det går att ändra på CSS, det är rekommenderat att göra det i temat istället för på den här sidan så för det mesta är det bara att gå vidare till nästa sida. Nu kommer den intressanta biten där jag väljer vad som skall synas i de två kolumnerna.

→ [Substitutions](#)



Figur 29. Layout i Page Manager.

Figur 29 visar hur det ser ut när jag satt till de fält jag ha vill till kolumnerna. För att göra det klickar jag på kugghjulet i övre vänstra hörnet och väljer Add content. Det kommer upp en ruta där jag väljer Nod och Field: Brödtext (body) för den vänstra kolumnen och Field: bild (field\_bild) för den högra kolumnen. Figur 30 visar hur det ser ut.



Figur 30. Lägg till fält att visa i kolumnerna.

För att bilderna skall visas i rätt format väljer jag som image style stilen *home\_thumb* som jag skapat själv. Stilen gör så att alla bilder kommer att visas i formatet 150px x 150px oberoende hur stor bilden var från början. Media stilar skapas i media styles under Configuration/media i administrationen.

När man klickar Add variant är layouten färdig bara man sparar. Nu kommer alla noder av content typen Textbild att formateras så att texten kommer i den vänstra kolumnen och bilderna i den högra. Automatiskt är kolumnerna jämbreda så jag har använt CSS för att göra dem så breda jag vill.

Panels layout är inte automatiskt responsiv så för att göra den responsiv lägger jag till den koden i kodexempel 10 under @media all and (max-width: 659px):

**Kodexempel 10. Kod för att göra panels kolumner responsiva.**

```
@media all and (max-width: 659px) {
  .panel-2col .panel-col-first {
    float: left;
    width: 100%;
  }

  .panel-2col .panel-col-first {
    float: left;
    width: 100%;
  }
}
```



**Figur 31. Hemsidan i Drupal.**

Figur 31 visar hemsidan i originalformat till höger och i mobilformat till vänster. Som syns flyttas bilderna under texten i mobilformatet.

## 12.2 Advanced Custom Field i WordPress

Advanced Custom fields (ACF) är en gratis insticksmodul till WordPress som hjälper dig att få en bättre användarvänlighet till pages och posts, med det menas att med ACF kan man lägga till mera flexibel data till noderna.

Man börjar med att laga en ny field group, i detta fall gav jag den namnet sidbild eftersom bilden skulle synas på sidor och inte posts. Sidbild är då fieldgruppens namn. I fieldgruppen sidbild lagade jag två nya fält som heter kuva och kuva1, deras fältnamn blev sidbild och sidbild1. För att fältet skall visas som en bild måste man lägga fältets typ till Image. Bildens storlek lagade jag till thumbnail vilket är minsta möjliga av de 4 alternativen, 150px x 150px. Sedan valde jag att fältets returvärde skulle vara som ID, vilket är viktigt att veta i senare skede när man med PHP kodar för att få bilderna att synas. Det går jag in på senare.

Edit Field Group
Add New

Sidbild ← Fältnamnet

| Field Order | Field Label | Field Name |
|-------------|-------------|------------|
| 1           | kuva        | sidbild    |

**Field Label \***

This is the name which will appear on the EDIT page

kuva

**Field Name \***

Single word, no spaces. Underscores and dashes allowed

sidbild ← Fältnamn

**Field Type \***

Image ← Fältets typ

**Field Instructions**

Instructions for authors. Shown when submitting data

**Required?**

Yes  No

**Return Value**

Specify the returned value on front end

Image Object  Image URL  Image ID ← Returvärdet

**Preview Size**

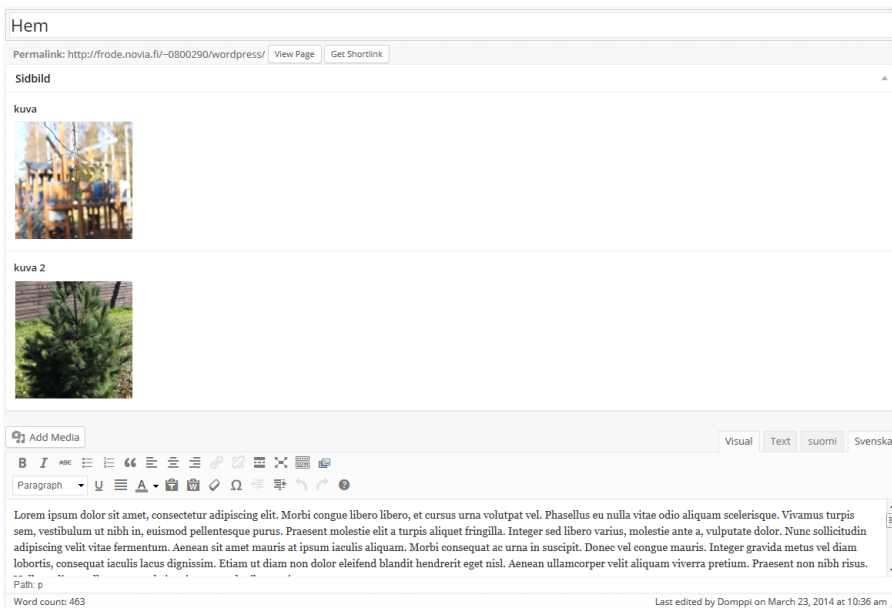
Shown when entering data

Thumbnail  Medium  Large  Full ← Bildstorlek

**Figur 32. Inställningarna för en Field Group.**

De här inställningarna som syns i figur 32 kommer att visas nästa gång jag lagar en ny sida. Ovanför texten kommer det att finnas två rutor som man kan lägga in en bild i. Figur 33 på nästa sida visar exempel på hur de ser ut när man skall laga en sida.





Figur 33. Exempel på hur det ser ut när man lagar en ny sida.

I bilden ser man att man kan lägga till två stycken bilder ovanför texten. De här bilderna kommer dock inte ännu att synas när man publicerar sidan, utan man måste först koda lite i PHP för att få det att fungera. PHP koden ser ut såhär i template filen.

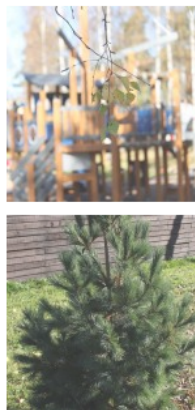
#### Kodexempel 11. PHP för att få bilderna att synas på sidan.

```
<div id="sidbild">
    <!-- kuva -->
    <?php $image = wp_get_attachment_image_src(get_field('sidbild'), 'thumbnail'); ?>
    <?php if($image[0]) { ?>
    
    <?php } ?>
    <!-- kuva1 -->
    <?php $image = wp_get_attachment_image_src(get_field('sidbild1'), 'thumbnail'); ?>
    <?php if($image[0]) { ?>
    
    <?php } ?>
</div>
```

Här ser man i koden att den visar bilderna genom funktionen *get\_field* som kallar fältnamnen *sidbild* och *sidbild1* och deras storlek kommer att visas som thumbnail på sidan. Alla i bilder är inne i regionen *sidbild* som är definerad som *float:right* i css filen vilket gör att bilden flyter till höger om texten på sidan.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi congue libero libero, et cursus urna volutpat vel. Phasellus eu nulla vitae odio aliquam scelerisque. Vivamus turpis sem, vestibulum ut nibh in, euismod pellentesque purus. Praesent molestie elit a turpis aliquet fringilla. Integer sed libero varius, molestie ante a, vulputate dolor. Nunc sollicitudin adipiscing velit vitae fermentum. Aenean sit amet mauris at ipsum iaculis aliquam. Morbi consequat ac urna in suscipit. Donec vel congue mauris. Integer gravida metus vel diam lobortis, consequat iaculis lacus dignissim. Etiam ut diam non dolor eleifend blandit hendrerit eget nisl. Aenean ullamcorper velit aliquam viverra pretium. Praesent non nibh risus. Nulla ut diam pellentesque, ultrices ipsum at, dapibus orci.

Donec imperdiet et orci a molestie. In diam elit, placerat a euismod non, luctus non sem. Aenean porta molestie nunc, sit amet condimentum nunc dignissim vitae. Fusce in venenatis nibh, sed luctus libero. Curabitur elementum sit amet enim ac molestie. Praesent vulputate sed libero in tempor. Duis at magna facilisis, porttitor mauris ut, blandit mauris. Sed egestas elit metus, quis accumsan ante gravida eget. Sed a mi eget sem ullamcorper tincidunt. Pellentesque ante lorem, adipiscing sit amet felis vitae, viverra commodo diam. Integer tempus urna ut est interdum fermentum. Proin ac nulla quam. Mauris ullamcorper sollicitudin sollicitudin. Donec odio eros, mattis ac pulvinar vel, laoreet et turpis. Suspendisse metus enim, luctus at tincidunt a, interdum a justo.



**Figur 34. Sidans slutliga utseende.**

I figur 34 ser man hur sidan ser ut när man har lagt in koden i php filen. Här ser man hur bilderna är på högra sidan om texten.

För att bilderna skall flyttas under texten i mindre skärmupplösning ändrar jag *sidbild* i CSS till 100% när @media regeln är max-width 650px. Bilderna flyter under texten eftersom de kommer under texten i template filen.

## 13 Senaste nyheter

Vi ville skapa en funktion som visar senaste nyheterna i sidfältet. Det kopplas också till en sida där den senaste nyheten visas. Vi har använt lite olika metoder för att få det här att fungera men slutresultatet är det samma.

Skillnaden mellan Drupal och WordPress är att arbetsmängden är lite mindre i WordPress och det är lite lättare om man inte har använt Views tidigare.

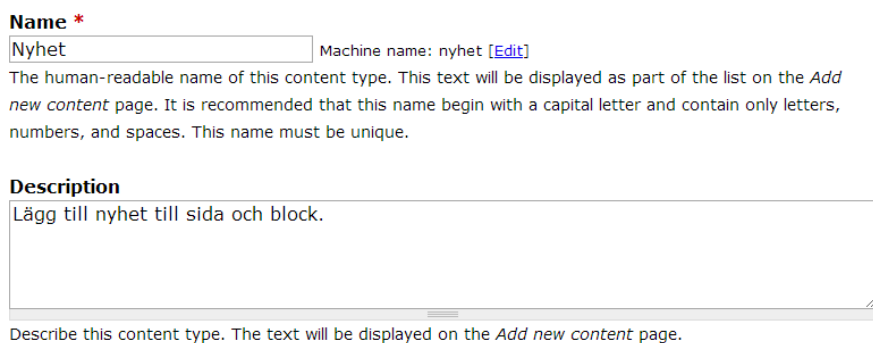
### 13.1 Senaste nyheter i Drupal

För att skapa ett block och en sida som visar de senaste nyheterna använde jag mig av insticksmodulen Views. Views är en av de mest använda och nyttiga modulerna som finns

till Drupal. Views kan användas till en mängd olika ändamål men i grunden är Views ett verktyg för att hämta data, behandla data och visa upp data.

För en Drupalutvecklare är Views antagligen den modul som det är viktigast att ha goda kunskaper i att använda. Jag kommer härnäst att förklara hur jag använt Views för att skapa ett nyhetsblock och en nyhetssida. Genom att följa samma steg går det att skapa i princip en hurudan vy som helst.

Efter att ha installerat Views, som vilken annan modul som helst, är det första steget att skapa en ny Content Type. Jag skapar en Content Type kallad *Nyhet* eftersom jag med hjälp av Views vill lista alla de senaste posterna från *Nyhet*. Det skulle gå att skapa en vy som listar alla poster men då skulle också huvudsidorna bli listade som nyheter.



**Name \***  
Nyhet Machine name: nyhet [\[Edit\]](#)

The human-readable name of this content type. This text will be displayed as part of the list on the *Add new content* page. It is recommended that this name begin with a capital letter and contain only letters, numbers, and spaces. This name must be unique.

**Description**  
Lägg till nyhet till sida och block.

Describe this content type. The text will be displayed on the *Add new content* page.

#### Figur 35. Skapa ny Content typ.

För att skapa en ny Content Type går man till Add Content Type som finns under Content Type i Structure. Det enda man behöver göra här är att ge ett namn och spara. Om man vill kan man skriva en beskrivning på vad Content Typen gör.

Nu är det dags att skapa vyn i Views. Det gör man under View i Structure.

**Add new view**

**View name \***  
 Nyheter  Machine name: nyheter [\[Edit\]](#)

Description

---

**Show** Content  **of type** Nyhet  **sorted by**  
 Newest first

---

Create a page

Page title

Path  
 http://127.0.0.1/drupal/sv

Display format  
 Unformatted list  of full posts   
 without links  without comments

Items to display

Create a block

Block title

Display format  
 Unformatted list  of titles (linked)

Items per page

Use a pager

**Figur 36. Skapa ny vy.**

I figur 36 syns inställningarna jag har för att skapa nyhetsvyn. Det viktiga är namnet och att *Show* är inställt som *Fields* och *of type* som *Nyhet*. Det här betyder att vyn kommer att visa fält från poster av Content Typen Nyhet som just skapades. I bilden syns också att jag har ikryssat *Create a block* för att skapa ett nyhetsblock.

## Displays

Page
Block
+ Add

▼ **Page details**

Display name: [Page](#)

**Title**

Title: [Nyheter](#)

**Format**

Format: [Unformatted list](#) | [Settings](#)

Show: [Fields](#) | [Settings](#)

**Fields** [Add](#) ▼

[Content: Title](#)

[Content: Body](#)

**Filter criteria** [Add](#) ▼

[Content: Published \(Yes\)](#)

[Content: Type \(= Nyhet\)](#)

**Sort criteria** [Add](#) ▼

[Content: Post date \(desc\)](#)

**Page settings**

Path: [/nyheter](#)

Menu: [Normal: Nyheter](#)

Access: [Permission](#) | [View published content](#)

**Header** [Add](#)

**Footer** [Add](#)

**Pager**

Use pager:

[Display a specified number of items](#)

| [1 item](#)

More link: [No](#)

Figur 37. Editering av vy.

Figur 37 visar hur editeringen av vyn Nyheter ser ut när den är färdig. Bara några små förändringar har gjorts gentemot en helt ny vy. Jag har satt till fältet *Content: Body* och ställt in så att sidan visas i menyn. Som syns i bilden på nästa sida har jag också satt till fältet *Content: Post date* till blocket och editerat fältet så att datumet visas som det skall i Finland. Det här illustreras i figur 38.

Härefter förklarar jag vad som editeras under de olika rubrikerna:

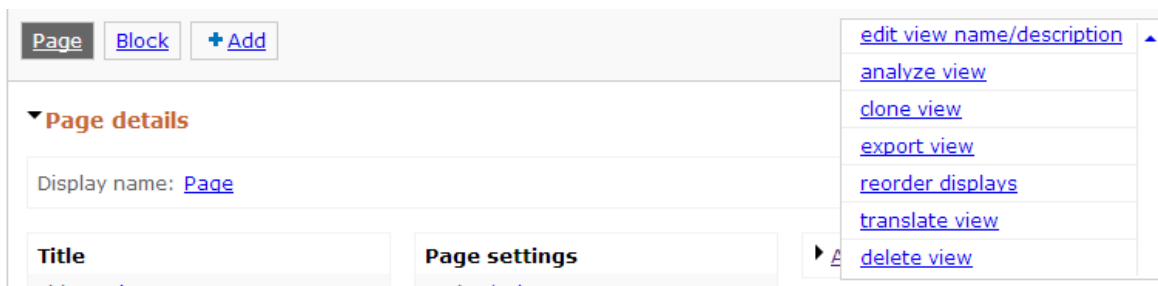
- Title: Titeln på vyn.
- Format: Det går att formatera vyn på olika sätt. T.ex. i en lista eller i en tabell.
- Fields: Vilka fält som skall visas i vyn.

- Filter criteria: Vyn kan filtreras enligt önskan. I det här fallet är posterna filtrerade så att de måste vara markerade som *Published* och av typen Nyhet.
- Sort criteria: Hur vyn skall sorteras
- Page settings: Här ställs sidans sökväg och meny in. Det går också att bestämma vem som skall ha tillgång till att se vyn.
- Header: Det går att sätta till någonting till sidans headerområde.
- Footer: Det går att sätta till någonting till sidans footerområde.
- Pager: Här specificeras hur många poster som skall synas i vyn och vad som händer om alla poster inte rymms.

The screenshot shows the configuration interface for a date field in a Drupal block. On the left, the 'Block detail' sidebar is visible, showing the title 'Nyheter', format 'Uniform', and fields 'Content: Title', 'Content: Post', and 'Content: Body'. The main configuration area on the right includes a description 'The date the content was posted.', checkboxes for 'Create a label' and 'Exclude from display', and a 'Date format' section with a dropdown menu set to 'Custom' and a text input field containing 'F j, Y'.

Figur 38. Postdatum i blocket.

För att översätta vyn använde jag tillägsmodulen Views Translate. Översättningen skapas genom att klicka på *translate view* i rullgarinsmenyn uppe till höger i vynes editeringsfönster. Där går det att översätta allting i vyn till de språk som är installerade på webbplatsen. Detta syns i figur 39 på nästa sida.

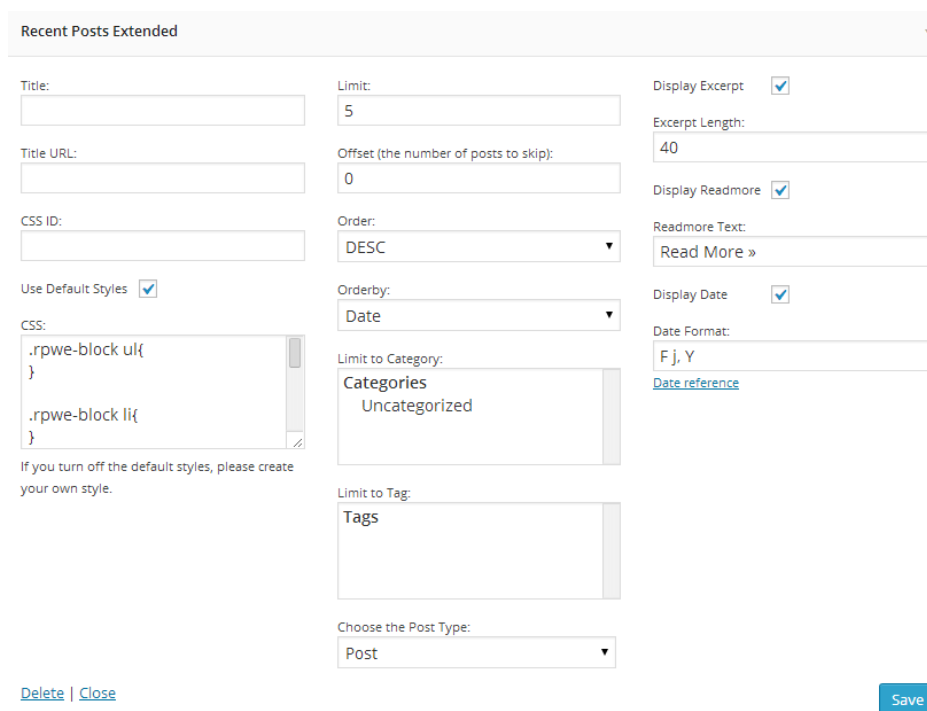


Figur 39. Översätta vy.

## 13.2 Senaste nyheter i WordPress

För att få senaste nyheterna att visas på sidfältet på webbplatsen använde jag mig av insticksmodulen Recent Posts Widget Extended. För att aktivera modulen måste man flytta över den till sidebaren i widget panelen.

Den här modulen fungerar så att varje gång du lagar en ny nyhet/post till webbplatsen så visas rubriken och lite text av ämnet i sidofältet på webbplatsen. Den visar också vilket datum nyheten är gjord. För att läsa hela nyheten kan man klicka på Read more eller på rubriken. Den här modulens funktioner behövde jag inte redigera med egen kod. Designen på rubrikerna och texten redigerade jag dock i egen css.



Figur 40. Inställningar för Recent Posts Widget Extended.

I redigeringsfältet kan man välja ganska mycket olika funktioner, så som t.ex. vilken ordning nyheterna kommer och datum. Sedan kan man också välja hur lång texten skall vara eller om man alls vill visa texten under rubrikerna. Det finns också css editor med i redigeringsfältet men jag tyckte det var lättare att ändra utseendet i min egen css fil.

Eftersom vi också skulle ha en sida som visar den nyaste nyheten så gjorde jag en egen template för nyhetssidan. I den här templatens lade jag till en rad kod som tar den senaste posten och visar den.

**Kodexempel 12. Kod för att visa den senaste nyheten på nyhetssidan.**

```
<?php query_posts ('category_name=senastenyhet&showposts=1'); ?>
```

Koden fungerar så att den visar den senaste posten i kategorin ”senastenyhet”.

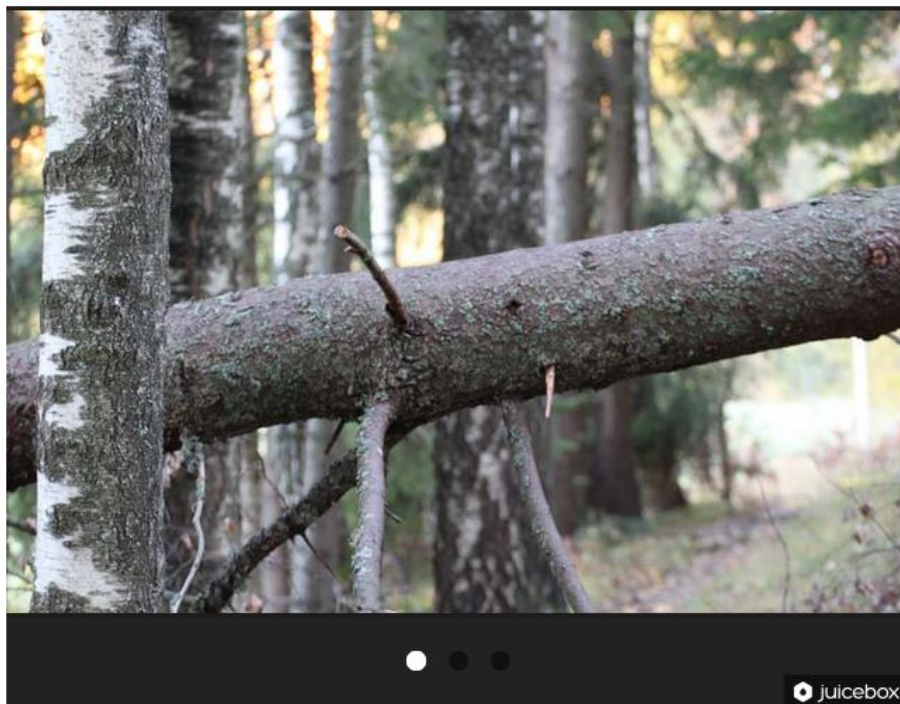
## 14 Galleri

Väldigt många webbplatser har någon sorts galleri. Därför bestämde vi oss för att webbplatsen vi skapade måste ha ett galleri. Vi testade flera moduler i båda CMS-systemen men de hade alla något problem, oftast med den responsiva layouten. Till sist hittade vi en insticksmodul som fungerar i både Drupal och WordPress i form av Juicebox.

Juicebox är ett populärt responsivt galleri. Gratisversionen av Juicebox har alla funktioner som behövs för en grundlayout. I betalversionen finns flera möjligheter till egen design och man slipper vattenmärket.



## Galleri



Figur 41. Juicebox galleriet.

### 14.1 Galleri i Drupal

För att använda Juicebox i Drupal krävs:

- Drupal Libraries API 2.0 eller högre
- Juicebox javascript bibliotek som finns att ladda ner från [www.juicebox.net/download](http://www.juicebox.net/download)
- Juicebox modulen för Drupal

Drupal Libraries API 2.0 och Juicebox modulen installeras som vanligt. Juicebox javascript bibliotek kommer i en zip-fil i vilken det finns en katalog med namnet *jbcore*. Den katalogen skall laddas upp till webbservern i `/sites/all/libraries/juicebox`. Sista steget i att installera Juicebox är att aktivera modulerna Drupal Libraries och Juicebox.

För att skapa ett galleri började jag med att skapa en ny content typ med namnet Gallery juicebox. Till den nya content typen lade jag till ett fält av typen Image med namnet `Imag_juicebox`. Det här fältet gör det möjligt att ladda upp bilder.

För att bilderna skall visas som ett Juicebox galleri skall man gå till Manage Display i content typen Gallery juicebox och välja Juicebox Gallery under Format för fältet Imag\_juicebox. Här går det att ställa in hur galleriet skall se ut.

| Field   | Label    | Format           |
|---|----------|------------------|
| + Body  | <Hidden> | Default          |
| + Imag_juicebox   | <Hidden> | Juicebox Gallery |
| Image style: Original Image<br>Thumbnail style: Juicebox square thumbnail (85x85)<br>Caption source: None<br>Title source: None<br>Additional Juicebox library configuration options may also be set. |          |                  |
| <b>Hidden</b>   |          |                  |
| + Title   | Above    | <Hidden>         |

Figur 42. Juicebox Gallery under Manage Display i Drupal.

När man nu skapar en ny nod av typen Gallery juicebox kommer den automatiskt att visas som ett Juicebox galleri.

## 14.2 Galleri i WordPress

Jag började använda WordPress kändaste gallery modul som heter NextGEN-gallery. Den här insticksmodulen är gratis men för att få bättre funktionalitet och flera layout alternativ så måste man köpa NextGEN-gallery pro.

För att sätta in ett galleri till modulen NextGEN-gallery går man till add new gallery/Images under rubriken gallery. Här väljer man sedan ett nytt galleri och börjar ladda upp bilder till det. Ditt färdiga galleri får ett id nummer vilket är väldigt viktigt att veta när man skall visa galleriet på webbplatsen.

NextGEN-gallery är responsivt, men eftersom jag inte fick den layouten jag hade önskat mig p.g.a. gratis versionens restriktioner började jag använda mig av modulen Juicebox.

För att få Juicebox att fungera i WordPress så måste man gå igenom samma installation som med andra insticksmoduler. När man väl har installerat och aktiverat modulen så finns det en färdig Add Juicebox ikon när du skriver en ny post. När du klickar på ikonen kommer det upp en lista med många olika alternativ. I listan kan du välja hur du vill att layouten skall se ut.

| Gallery Options        |                                     |                          |                                     |
|------------------------|-------------------------------------|--------------------------|-------------------------------------|
| Gallery Title          | <input type="text"/>                |                          |                                     |
| Image Source           | NextGEN Gallery ▾                   |                          |                                     |
| NextGEN Gallery Id     | <input type="text" value="22"/>     |                          |                                     |
| Gallery Width          | <input type="text" value="100%"/>   | Gallery Height           | <input type="text" value="600px"/>  |
| Background Color       | <input type="text" value="222222"/> | Background Opacity       | <input type="text" value="1"/>      |
| Text Color             | <input type="text" value="ffffff"/> | Text Opacity             | <input type="text" value="1"/>      |
| Thumbnail Frames Color | <input type="text" value="ffffff"/> | Thumbnail Frames Opacity | <input type="text" value="0.5"/>    |
| Show Open Button       | <input checked="" type="checkbox"/> | Show Thumbs Button       | <input checked="" type="checkbox"/> |
| Show Expand Button     | <input checked="" type="checkbox"/> | Use Fullscreen Expand    | <input type="checkbox"/>            |
| Use Thumbnail Dots     | <input type="checkbox"/>            |                          |                                     |

**Figur 43. Inställningar för Juicebox.**

Figur 43 visar hur att jag valt NextGEN Gallery som Image Source eftersom jag skapat ett galleri och laddat upp bilder med den modulen. Det här betyder att Juicebox tar över ett färdigt gjort NextGEN galleri. Jag har valt Gallery Id 22 eftersom jag har ett galleri med den id:n gjord. Juicebox fungerar i princip bara som ett tema för hur bilderna visas på webbplatsen. Det behövs inte heller användning av egen CSS kod eftersom det finns tillräckligt med design alternativ i redigeringsfältet.

## 15 Intervju

För att få inblick i hur CMS-system används professionellt i ett riktigt företag har vi intervjuat grundaren av företaget Codes Digital Media. Vi har också intervjuat Mathias Bilenberg som använt båda systemen men som nu specialiserat sig på Drupal. Intervjuerna följer den kvalitativa forskningsmetoden eftersom vi har intervjuat bara två personer. Vi använder intervjuerna för att stöda våra egna tankar då vi studerar skillnaderna mellan Drupal och WordPress.

### 15.1 Kvalitativ metod

Kvalitativ metod använder man när man vill få svar på frågor som det inte går att mäta genom absoluta data. I kvalitativa studier utgår man ifrån att verkligheten kan uppfattas på olika sätt och det finns inget entydigt svar. Eftersom det inte riktigt går att mäta att Drupal eller WordPress absolut är bättre eller sämre så har vi ställt frågor enligt den kvalitativa

metoden. Den kvalitativa processen är induktiv vilket betyder att man försöker komma fram till en förklaring på basis av sina iakttagelser eftersom det inte går att göra exakta mätningar som kan bevisas vara sanna. Den kvalitativa metoden är självklart mera subjektiv än den kvantitativa metoden men det går inte att göra någonting åt det eftersom det är omöjligt att mäta objektivt vad som är bättre och sämre mellan Drupal och WordPress. I vissa fall kan någonting vara bättre för en specifik lösning men sämre för en annan.

Vår intervju är en *semistrukturerad intervju* vilket betyder att man frågar mera öppna frågor istället för detaljerade frågor med entydiga svar. Idén är att den intervjuade skall få utveckla sina svar själv utan att ledas av intervjuaren. Till det här hör också att vi använt uppföljningsfrågor för att uppmuntra den intervjuade att utveckla sina svar.

## **15.2 Egenföretagare Mathias Bilenberg**

Mathias började arbeta med webbutveckling som hobby redan på 90-talet. CMS-system började han använda i början av 2000-talet i form av Joomla!. Efter Joomla! gick han över till WordPress och nu använder han främst Drupal. Under den senaste tiden har han främst arbetat med webbutveckling och internetmarknadsföring åt mindre företag i Västnyland. Några stora företag han har jobbat för med Drupalutveckling är Snellman och Mush. Härefter följer en intervju med Mathias Bilenberg.

## **15.3 Intervju med Mathias Bilenberg**

### **Vad tycker du om Drupal i jämförelse med Joomla! och WordPress?**

Jag använder idag hellre Drupal än andra CMS-system. Orsaken till det tror jag främst är att jag haft turen att arbeta med, och fått stöd av människor som kan systemet otroligt bra. Tekniska orsaker till att jag valt att använda Drupal är att det finns ett stort antal tilläggsmoduler som är relativt enkla att anpassa till egna behov. I Drupal är det också ganska enkelt att själv skapa moduler, även om man är en relativt ovan användare av CMS-systemet. Dessutom är Drupal-communityn väldigt stor och aktiv, och det finns alltid bra hjälp att få, både genom att läsa gamla frågeträdar, eller att ställa en fråga i communityn.

### **Vad finns det för fördelar och nackdelar med Drupal?**

Jag skulle säga att möjligheterna att anpassa Drupal till just sina egna behov är den största fördelen Drupal har till sina konkurrenter. Samtidigt är det också en nackdel om man inte är en van användare av CMS-systemet. Det går t.ex. mycket snabbare att komma igång med ett CMS-system som WordPress än Drupal, men för att anpassa vissa delar av CMS-systemet krävs betydligt mycket mera jobb i WP än i Drupal. Jag skulle också säga att saknaden av ett klart ”back-end”-interface i Drupal är en **FÖRDEL**, eftersom man så pass enkelt kan anpassa hur drupal skall se ut, och användas, av olika användartyper.

**Enligt statistiken [http://w3techs.com/technologies/market/content\\_management](http://w3techs.com/technologies/market/content_management) så används Drupal på större och mera trafikerade sidor än andra CMS-system. Har du någon uppfattning om vad det beror på?**

Jag skulle säga att Drupals anpassningsbarhet och flexibilitet är det som gör att man väljer att bygga större och mer komplexa sidor i Drupal istället för andra CMS-system. Det är klart att system som WordPress kan användas på större sidor med många användare, ett exempel på det är New York Times.

**Använder du mest färdiga moduler i Drupal eller har du kodat mycket själv?**

Största delen av modulerna jag använder är färdiga Drupal-moduler, och det jag gillar med Drupal är att det går enkelt och det finns mycket hjälp att få i Communityn på det området. Drupal är också smidigt när det gäller Moduler så att det går relativt enkelt att skapa små egna moduler för att göra enkla grejer. T.ex. för en så enkel grej som att köra någon typ av JavaScript på olika ställen på sidan.

**Om du har kodat själv, vad beror det på?**

Oftast handlar det om att man vill anpassa någon färdig modul lite för eget behov, eller skriva någon kort modul som bygger på någon annan modul. T.ex. ändra på sättet som webform visar tokens, i8n hanterar val av språk eller hur views hanterar språkfilter.

**Vi såg på din Facebook-status att du har sett ljuset när det gäller Drupal, vad fick dig att göra det?**

Jag arbetar på ett lite större multisite-system och konstaterade att jag gillar sättet som olika moduler fungerar ihop, hur bra helheten fungerar och hur enkelt det går att anpassa Drupal för mina egna behov.

## **15.4 Codes Digital Media**

Codes Digital Media är ett företag som arbetar med Webbutveckling, innehållshantering och webbhotell. De är specialiserade på att använda WordPress. Företaget består av Conny Karlsson och Celinda Byskata. Conny arbetar som webbutvecklare medan Celinda har hand om innehållet på webbplatsen. Företaget samarbetar med J&CO och Lara-X som har kontor i samma byggnad. Största webbplatsen som Codes har skräddarsytt är *anna.fi* och de har också arbetat med projektet *kaksplus.fi*. Livligaste använda Webbshoppen de har utvecklat är sidan *rengasajatarvike.com/verkkokauppa*.

Nedan kommer en intervju med Codes Digital Medias grundare och webbutvecklare Conny Karlsson.

## **15.5 Intervju med Codes Digital Medias grundare Conny Karlsson**

### **Har du erfarenhet av andra CMS system än WordPress?**

Jag har lite erfarenhet av Joomla, men det är redan många år sedan jag gjort något med det. Dessutom har jag jobbat ganska mycket med vBulletin, som egentligen är ett diskussionsforum, men som med tilläggsmoduler har gjorts om till ett CMS.

### **Ifall du har så vad tycker du om WordPress jämfört med dem?**

Wordpress är väldigt populärt och det finns en massa utvecklare som gör moduler samt förnyar själva CMS:et vilket gör att det är smidigt att använda, och alltid är up to date. Det är dessutom enkelt att använda både som utvecklare och slutanvändare. Jag valde i tiderna

att inte använda de andra CMS systemen för att de verkade för krångliga både att arbeta med, men också för slutkunden.

### **Varför gick du över från ett eget CMS system till WordPress?**

Jag byggde ett eget CMS i början av 2004 för att jag då inte hittade ett lättanvänt CMS på marknaden som jag skulle ha kunnat använda mig av. Jag ville ha ett simpelt verktyg med lätta (och få) funktioner, som kunden lätt klarar av att använda själv. Dessutom ville jag då lära mig att programmera i PHP och MySQL, så att bygga ett CMS var ett ypperligt sätt att så att säga slå två flugor i en smäll. Jag har nu sedan 2012 använt enbart Wordpress. Jag kom i kontakt med det av en slump och konstaterade att det är lätt att använda och arbeta med och att många idag använder sig av just det verktyget. Problemet med mitt eget CMS var att det hade fått en hel del funktioner, som med tiden blev föråldrade, vilket betydde att en stor del av min tid gick åt till att gratis uppdatera det så att det skulle fungera med dagens kvar och standarder. Wordpress utvecklas hela tiden av andra utvecklare, så där behöver man inte själv sätta ner tid och resurser på att hålla programmet up to date, och skall man göra något nytt, finns det högst antagligen någon som gjort något liknande tidigare, som man kan dra nytta av, jämfört med att med mitt egna CMS behöva bygga upp det från början.

### **Vad finns det för fördelar med WordPress?**

Lättanvändligheten i att utveckla CMS till allt från enkla system till mycket avancerade.

### **Vad finns det för nackdelar med WordPress?**

Inget vad jag kommer på.

### **Finns det bra utbud på gratis insticksmoduler till WordPress till allt du behöver?**

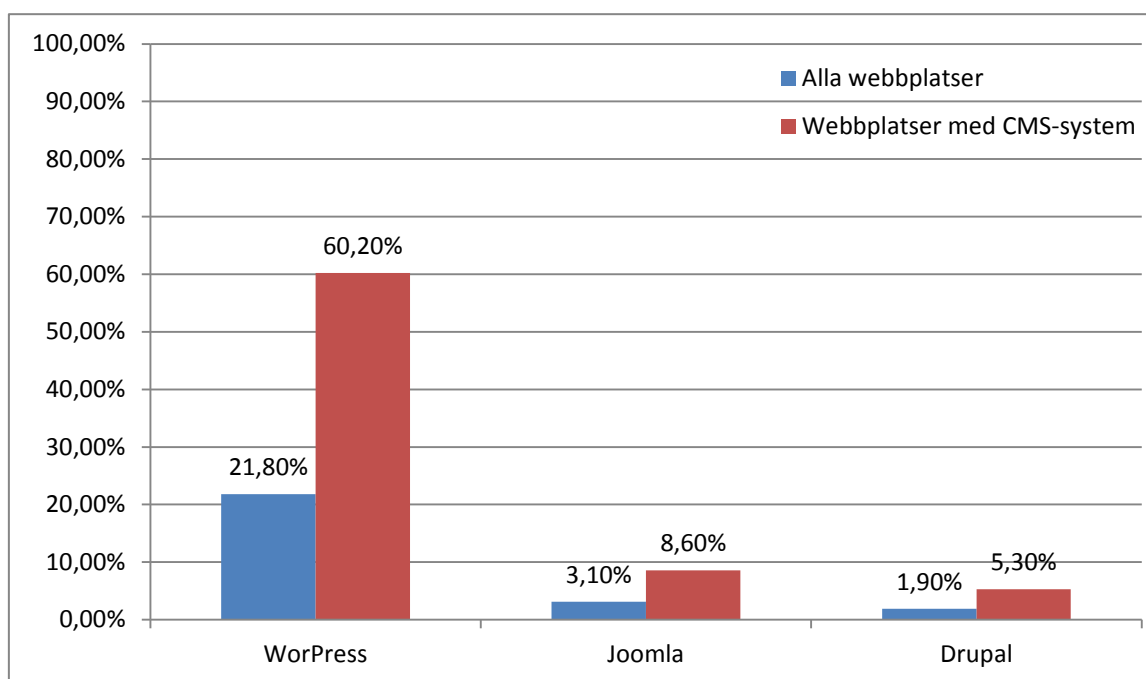
Det finns en hel del moduler som jag använder, varav många har en gratisversion som man klarar sig med i de flesta fall, men för en liten slant kan man köpa sig en sk pro-version av en modul om man behöver.

### Använder du moduler som kostar?

Jag har köpt en så kallad utvecklarcens till några moduler, så att jag kan använda dem i alla mina projekt. Det är moduler som jag använder i 99% av hemsidor jag gör, vilket betyder att det lönar sig att ha den typen av licens. Till vissa projekt har jag köpt en engångslicens till någon modul. Då är det frågan om någon typ av speciallösning som jag kan anta att jag inte behöver någon fler gång. Prissättningen på dessa moduler tycker jag är helt human, och man kan lita på att de är gjorda så att de verkligen fungerar. På gratismarknaden kan det däremot finnas moduler som inte alltid är kompatibla med alla andra funktioner, vilket ibland kan vålla problem. Men med tiden lär man sig vilka som går att använda och vilka som inte fungerar.

## 16 Skillnader mellan Drupal och WordPress

WordPress är det mest använda CMS-systemet i världen och Drupal är det tredje mest använda.

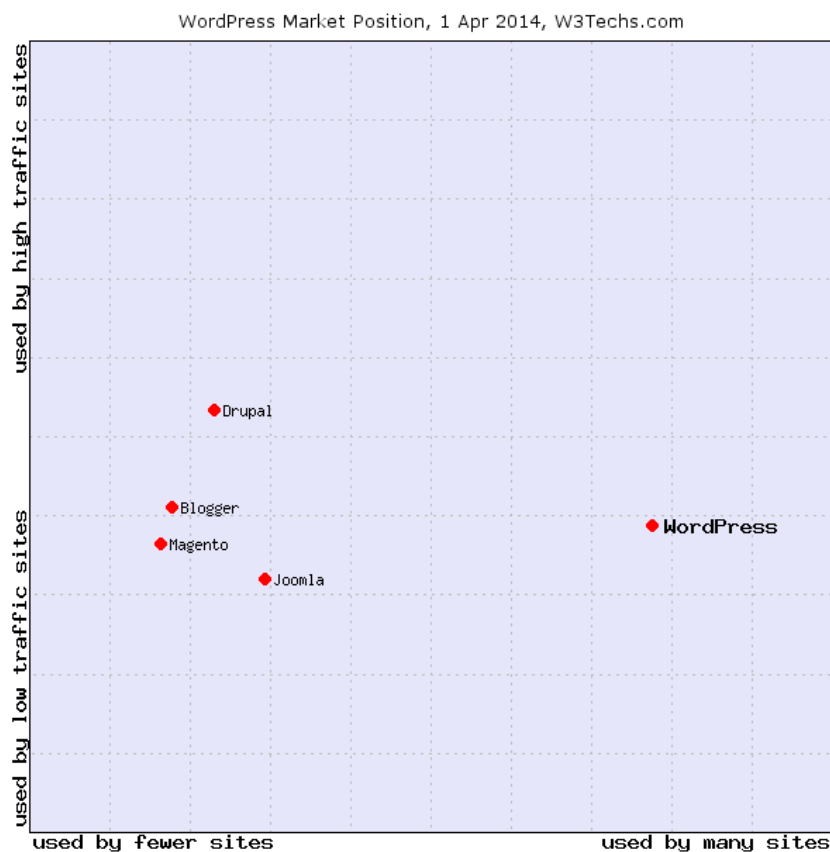


Tabell 1. CMS-systemens marknadsandel den 1 april 2014.



Tabell 1 visar att WordPress är det överlägset mest använda CMS-systemet med 60% av alla webbplatser gjorda med ett CMS-system. Det som sticker ut är att 21,8% av alla webbplatser på hela internet är byggda med WordPress. Orsaken att WordPress är så dominerande är WordPress.com som är en av de största blogg tjänsterna i världen (W3techs 2014).

Det här säger inte allt om CMS-systemens marknadsposition. Trots att WordPress används av överlägset flest webbplatser så använder många webbplatser med mycket trafik Drupal. De flesta källorna vi hittade sade att Drupal är det bästa verktyget för större webbplatser medan WordPress är bra för små webbplatser. Vi hittade inte en bra förklaring på varför det skulle vara så nu mera. WordPress har haft ett rykte av att vara bara ett bloggverktyg men det uppdateras ofta och har redan i flera år varit ett komplett CMS-system som det går att göra vilken sorts webbplats som helst med.



**Tabell 2. CMS-systemens marknadsposition.**

Som Tabell 2 illustrerar är WordPress använt av överlägset flest webbplatser men Drupal används på webbplatser som har mycket trafik. Den största är *twitter.com* (W3techs 2014).

Målet med arbetet var att jämföra uppbyggnaden av en webbplats med Drupal och WordPress. Med det menar vi dels skillnader i moduler och installation, dels användarvänlighet. Funktionerna på webbplatsen har vi tagit upp tidigare i arbetet så här kommer vi att ta upp skillnaderna i systemen.

Drupal och WordPress har en nästan identisk installation. Den är inte helt enkel för en nybörjare men det finns tydliga instruktioner på CMS-systemens webbplatser. Själva byggandet av webbplatsen är ganska likartad men layouten för administration är mycket tydligare och mindre komplicerad i WordPress. Det här betyder att det utan tidigare erfarenhet är lättare att komma igång med WordPress än med Drupal. Eftersom vi är insatta i båda CMS-systemen från förut så var arbetet ungefär lika komplicerat och tidskrävande för båda webbplatserna men för en nybörjare skulle det ta mycket mera tid att bygga en webbplats med Drupal.

Problemet med Drupals administration är att alla moduler inte följer samma logik. T.ex. skulle det vara logiskt om allt på webbplatsen som har med media att göra skulle administreras under menyn media. Så är det inte, vissa moduler skapar helt egna administrationsmenyer och det tar tid att lära sig var allting finns.

Som Mathias Bilenberg påpekar kan Drupals administration också vara en fördel eftersom det går att anpassa "back-enden" helt som kunden vill. Det här är en mycket bra funktion men det kräver betydligt mera jobb av webbutvecklaren.

WordPress administrationssida har däremot en klar och tydlig navigering där allt följer samma logik. Som Conny Karlsson sade betyder det här att det är lättare för webbutvecklaren men **speciellt** för slutanvändare d.v.s. kunden att själv lägga till innehåll på webbplatsen.

Självklart har vi använt oss av olika lösningar för att skapa funktionerna på webbplatsen. För de flesta funktionerna var arbetsmängden lika stor. Den enda stora skillnaden var flerspråksfunktionen som kräver mycket mera tid och arbete i Drupal.

Man märker en stor skillnad i var tyngdpunkten ligger för Drupal och WordPress i hur deras support ser ut. Man kan hitta lösningar på de flesta vanliga problem för båda systemen men Drupals support räknar med att man har en viss teknisk förkunskap.

WordPress har däremot användare alltifrån bloggare till erfarna webbutvecklare vilket betyder att deras support måste vara nybörjarvänlig. P.g.a. det här lönar det sig för en nybörjare att börja med WordPress.

## 17 Utvecklingsmöjligheter

Det går att utveckla webbplatsen vi skapat i vilken riktning som helst. Den är bara en väldigt bra grund till en mera avancerad webbplats. Funktionerna på webbplatsen går också att utveckla men de här fungerar som en bra bas.

Om man vill göra en mera noggrann granskning mellan olika CMS-system skulle det vara en god ide att också ta med Joomla eftersom det är det tredje stora CMS-systemet och det är till funktionen lite mittemellan Drupal och WordPress. Det här kom vi på först när vi redan var på god väg med arbetet. Det skulle ha krävt mera tid än vi hade på tu man hand.

Tanken med arbetet var att det skulle vara nybörjarvänligt och den statistiska jämförelsen är också väldigt grundläggande. P.g.a. nybörjarvänligheten blev arbetet redan nu ganska omfattande. För att göra en mera noggrann jämförelse skulle det krävas mera data från användare av de olika CMS-systemen.

## 18 Slutsatser

Det här arbetet fungerar som en introduktion till att bygga en webbplats med CMS-system dock är inte lösningarna vi använt till de olika funktionerna de enda möjliga lösningar. Det här beror på att det finns så många moduler att välja mellan och självklart går det också att skapa helt egna funktioner. Eftersom vi använt oss av väldigt populära moduler så tycker vi att det här arbetet fungera som en bra introduktion till webbutveckling med CMS-system.

Ett av våra huvudmål var att lära oss att skapa en helt responsiv webbplats med hjälp av CMS-system. Det intressanta var att få alla moduler att fungera responsivt. Arbetsprocessen var mycket lärrik och nu vet vi hur man gör en responsiv webbplats med CMS-system.

Intervjuerna sammanföll ganska bra med våra egna tankar. Som vi har konstaterat är funktionaliteten hos båda systemen liknande men WordPress är ett mera nybörjarvänligt system. Om man skall skapa en enkel webbplats och överlämna den åt en kund som sköter om innehållet är det lättare med WordPress. Om man däremot skall skapa en större webbplats som man vill anpassa helt själv kan det vara en fördel att använda Drupal, förutsatt att man har erfarenhet av systemet.

## 19 Avslutning

Vi valde att göra det här arbetet tillsammans eftersom vi har tidigare erfarenhet av att arbeta tillsammans i form av en praktik. Vi funderade en tid på vad arbetet skulle handla om men vi kom på iden när Tomas Söderlund under praktiken hade jobbat med WordPress. Arbetsfördelningen var självklar på grund av det här.

Arbetet gick oväntat smidigt med få problem. Det enda vi hade en hel del problem med var galleriet, vilket troligen berodde på restriktioner på webbservern. Vi planerade arbetet ganska noggrant före vi alls började och lade upp en tidtabell som vi lyckades följa nästan helt och hållet. Den praktiska delen tog längre än väntat eftersom vi testade många olika insticksmoduler till de olika funktionerna. Det här berodde till stor del på att många moduler inte fungerar med responsiv design.

När vi började arbetet var planen att det skulle vara en jämförelse mellan de olika funktionerna och modulerna i Drupal och WordPress men medan vi arbetade kom vi fram till att arbetet skulle kunna fungera bra som en slags manual för nybörjare. Slutligen blev det en ganska jämn fördelning mellan de här båda delarna.

Arbetet var givande och roligt. Fastän vi båda hade använt oss av CMS-systemen förut så lärde vi oss nya intressanta funktioner eftersom vi testade olika alternativ. Det mest nyttiga och intressanta som vi lärt oss mera om är responsiv design. Det hade vi inte alls fått någon utbildning om i studierna eftersom det är så nytt vilket betyder att vi lärt oss allt på egen hand. Vi visste före vi började arbetet hur man gör en responsiv layout till webbplatsen men det var intressant att granska vilka moduler som fungerade responsivt och göra så att en del moduler fungerar responsivt som inte gör det automatiskt.

Målen med arbetet var att skapa en webbplats i Drupal och WordPress samt jämföra arbetsprocessen. Arbetet skulle också förklara arbetsprocessen på ett nybörjarvänligt sätt. Vi tycker att vi har uppnått de båda målen på ett bra sätt.

# KÄLLFÖRTECKNING

## LITTERATUR

Falk, J. 2011. *Drupal 7: Börja Här*. Lund: Studentlitteratur.

Gilmore, W.J. 2010. *Beginning PHP and MySQL: From Novice to Professional*. New York: Apress.

Korpela, J. 2003. *CSS-tyylit*. Porvoo: WS Bookwell.

Rantala, A. 2005. *Web-ohjelmointi*. Jyväskylä: Docendo.

Staflin, R. 2001. *Internet HTML- och XHTML boken*. Stockholm: Pagina.

## ELEKTRONISKA KÄLLOR

Ethan Marcotte (2014) <http://alistapart.com/article/responsive-web-design> [hämtat 31.3.2014]

W3techs (2014) [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all) [hämtat 31.3.2014]

Drupal (2014). History. <https://drupal.org/about/history> [hämtat 31.3.2014]

## FIGURFÖRTECKNING

|   |    |
|---|----|
| Figur 1. Bilden visar content och sidebar i fullskärm.          | 13 |
| Figur 2. Bilden visar webbplatsen som den ser ut i mobilformat. | 15 |
| Figur 3. Exempel på hur man skapar databas.                     | 16 |
| Figur 4. Filezilla när man installerar Drupal.                  | 17 |
| Figur 5. Filattribut i Filezilla.                               | 17 |
| Figur 6. Välj profil.   | 18 |
| Figur 7. Välj språk.  | 19 |
| Figur 8. Konfiguration av databas.                              | 19 |
| Figur 9. Drupal installerat.                                    | 20 |
| Figur 10. Nice Menus slutliga utseende.                         | 25 |
| Figur 11. Lista över vilka menyer som skall göras responsiva.   | 26 |
| Figur 12. Den responsiva menyn.                                 | 26 |
| Figur 13. Redigeringsprogrammets utseende.                      | 27 |
| Figur 14. Responsiva navigationen i WordPress.                  | 29 |
| Figur 15. Skillnaden mellan Content och Entity Translation.     | 31 |
| Figur 16. Language.   | 32 |
| Figur 17. Language Detection and Selection.                     | 32 |
| Figur 18. Flerspråkiga noder.                                   | 33 |
| Figur 19. Field translation.                                    | 33 |
| Figur 20. Replace titel.  | 33 |
| Figur 21. Lista över översättningar på en nod.                  | 34 |
| Figur 22. I bilden syns hur finska och svenska är valda.        | 35 |
| Figur 23. Inställning för URL val av språk.                     | 35 |
| Figur 24. Widgeten för språkval.                                | 35 |
| Figur 25. Page Manager i Drupal.                                | 37 |
| Figur 26. Add variant i Page Manager.                           | 37 |
| Figur 27. Selection rules i Page Manager.                       | 38 |
| Figur 28. Layout i Page Manager.                                | 38 |
| Figur 29. Layout i Page Manager.                                | 39 |
| Figur 30. Lägg till fält att visa i kolumnerna.                 | 39 |

|   |    |
|---|----|
| Figur 31. Hemsidan i Drupal.                                  | 40 |
| Figur 32. Inställningarna för en Field Group.                 | 41 |
| Figur 33. Exempel på hur det ser ut när man lagar en ny sida. | 42 |
| Figur 34. Sidans slutliga utseende.                           | 43 |
| Figur 35. Skapa ny Content typ.                               | 44 |
| Figur 36. Skapa ny vy.  | 45 |
| Figur 37. Editering av vy.                                    | 46 |
| Figur 38. Postdatum i blocket.                                | 47 |
| Figur 39. Översätta vy.                                       | 48 |
| Figur 40. Inställningar för Recent Posts Widget Extended.     | 48 |
| Figur 41. Juicebox galleriet.                                 | 50 |
| Figur 42. Juicebox Gallery under Manage Display i Drupal.     | 51 |
| Figur 43. Inställningar för Juicebox.                         | 52 |



## KODFÖRTECKNING

|   |    |
|---|----|
| Kodexempel 1. Kodexemplet visar hur responsiv design fungerar.      | 13 |
| Kodexempel 2. Kodexemplet visar hur @media regeln fungerar.         | 14 |
| Kodexempel 3. Aktivering av Widgets och sidfältet.                  | 24 |
| Kodexempel 4. Aktiveringskoden i templatefilen.                     | 24 |
| Kodexempel 5. Slutliga koden.                                       | 24 |
| Kodexempel 6. Kod för att placera in navigationsmodulen.            | 27 |
| Kodexempel 7. Kod för att visa den responsiva navigationen.         | 28 |
| Kodexempel 8. Kod som skapar en navigeringslista.                   | 29 |
| Kodexempel 9. PHP kod för att få widgetrubriken översatt.           | 36 |
| Kodexempel 10. Kod för att göra panels kolumner responsiva.         | 40 |
| Kodexempel 11. PHP för att få bilderna att synas på sidan.          | 42 |
| Kodexempel 12. Kod för att visa den senaste nyheten på nyhetssidan. | 49 |

## **TABELLFÖRTECKNING**

|   |    |
|---|----|
| Tabell 1. CMS-systemens marknadsandel den 1 april 2014. | 57 |
| Tabell 2. CMS-systemens marknadsposition.               | 58 |