

Responsiv webbutveckling med Drupal

Examensarbete

Medieteknik

2012

Patrik Storm

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Medieteknik
Identifikationsnummer:	3240
Författare:	Patrik Storm
Arbetets namn:	Responsiv webbutveckling med Drupal
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	
<p>Sammandrag:</p> <p>I dag används allt mera mobila verktyg, som smarttelefoner, surfplattor och små bärbara datorer, vilka har en gemensam egenskap; en liten skärm. Skärmstorleken ger krav på funktionaliteten på de webbplatser man surfar på, och detta problem har traditionellt lösts med att utveckla separata mobila nätsidor. Fast denna teknik är fungerande uppkommer ändå vissa problem. Då man har separata webbplatser måste båda uppdateras, och ifall någon delar länkar mellan olika medium kommer slutanvändaren att kanske använda sig av mobilversionen på en normalstorleks skärm. Dessa problem uppstår särskilt bland sociala media, då länkar delas ut bland mobila och persondatorer. Att lösa dessa problem har det utvecklats nya tekniker vilka möjliggör utvecklandet av en responsiv webbplats, som fungerar lika på alla medium, oberoende om skärmstorleken. Med nya egenskaper inom CSS3 kan man implementera sätt att ta reda på användarens skärmstorlek och dela information och media till användaren enligt användarens skärmstorlek. Denna teknik fungerar i de flesta webbläsare, som nyaste mobila verktyg har. Detta arbete koncentrerar sig på att implementera responsiva egenskaper till Drupal, m. ha. dagens nyaste och mest använda tekniker. Arbetet genomförs med olika tester och experiment, var resultatet testas med olika mobila verktyg, samt vanliga webbläsare. Arbetet begränsas till Linux AMP (LAMP) miljön och Drupal, men dessa tekniker kan användas och implementeras relativt enkelt till alla olika CMS system, eller helt skräddarsydda webbprojekt.</p>	
Nyckelord:	Drupal, Media Query, Omega ramverk, CSS3, Responsiv web-butveckling, mobila nätsidor, smarttelefon.
Sidantal:	50
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Media Technology
Identification number:	3240
Author:	Patrik Storm
Title:	Responsiv webbutveckling med Drupal
Supervisor (Arcada):	Johnny Biström
Commissioned by:	
<p>Abstract:</p> <p>There is a rise in the use of different smartphones, internet tablets and other portable devices capable of browsing the web. They all share one thing in common; a small screen size. The device screen size gives us some restrictions on the webpage or application we want to use. This problem has traditionally been solved by creating a totally separate webpage targeted for mobile-only devices. However, this technique is not without problems, when updating the these pages there will be extra work because the webpages are separate, with their own information. In other words each update must be done twice. A other aspect is the popularity among different social media, users tend to share links and information with both portable devices and personal computers. This results in smartphone users getting the mobile webpages links and vice versa. New technologies have been developed that makes it possible to create responsive webpages that work with any type of screen size. With new features introduced with the release of CSS3 it is now possible to make contextual queries to the users device and present the information and media accordingly. This new technology is supported by all the newest web browsers that smartphones are equipped with today. In this project there will be a study how these techniques can be imported to Drupal. The study includes tests and experiments done with smartphones, internet tablets and desktop browsers. The project is limited to the Linux platform running AMP (LAMP) and Drupal. However these techniques can be imported to other CMS systems, or customized web projects with little work.</p>	
Keywords:	Drupal, Media Query, Omega framework, CSS3, Responsive web-development, Mobile pages, Smartphone.
Number of pages:	50
Language:	Swedish
Date of acceptance:	

INNEHÅLL / CONTENTS

1	Inledning.....	10
2	Webbens utveckling	11
2.1	Syfte	12
2.2	Metoder	14
2.3	Avgränsning.....	14
2.4	Terminologi och förkortningar.....	15
3	Responsiv Webbutveckling	16
3.1	Tre viktiga elementen i responsiv webbutveckling	19
3.2	En flexibel, grid-baserad layout	19
3.3	Media queries	19
3.4	Media Query syntax.....	20
3.5	Flexibel media	23
4	Teknik.....	25
4.1	Drupal	25
4.2	Grid systemet	26
4.3	Grid baserad webbdesign	27
4.4	Pixel Grid eller Fluid Grid	30
4.5	CSS metoder	31
4.5.1	<i>Push- och pull CSS klasser.....</i>	<i>32</i>
4.5.2	<i>Prefix- och suffix CSS klasser.....</i>	<i>33</i>
5	Omega Ramverket	34
5.1	Organiserad CSS i Omega ramverket.....	35
5.2	Mobile First	36
5.3	CSS Strukturen.....	37
5.4	Omega administration och utveckling	43
6	Moduler.....	46
6.1	Context	46
6.2	Omega Tools	47
6.3	Chaos Tool Suite (Ctools)	47
6.4	Delta	47
6.5	Drush	48
7	Avslutning och diskussion	49
8	Resultat	49

Figurer

Figur 1. Google trends söknings resultat utan tidsbegränsning.....	15
Figur 2. Google trends söknings resultat för de senaste 12 månaderna.....	16
Figur 3. Demonstration av div element i ett Grid system.....	25
Figur 4. Demonstration på en 960 pixlars Grid med 60 pixlars kolumner.....	26
Figur 5. Demonstration av ett 16 kolumner brett Grid system.....	27
Figur 6. En demonstration över en webbplats med 12 kolumners Grid.....	28
Figur 7. Demonstration hur push och pull metoderna fungerar.....	30
Figur 8. Ett element i ett 12 kolumner brett Grid system.....	31
Figur 9. En demonstration över hur prefix och suffix metoder fungerar.....	31
Figur 10. Skärmstorleken är under 740 pixlar.....	36
Figur 11. Skärmstorleken är mellan 740 och 800 pixlar.....	38
Figur 12. Skärmstorleken är mellan 800 och 1024 pixlar.....	39
Figur 13. Skärmstorleken är större än 1220 pixlar.....	40
Figur 14. Omega ramverkets administration.....	43
Figur 15. Header sektionens konfiguration.....	44
Figur 16. Konfigurationen för de individuella regionerna.....	45

Tabeller

Tabell 1. Tabell med media query syntax.....	10
Tabell 2. Tabell av webbläsare som stöder media queries.....	19
Tabell 3. Tabell över media query syntax med olika värden.....	20

FÖRORD

När du läser detta arbete kan jag inte veta vilket medium du använder dig av. Kanske du har printat arbetet och läser direkt från papper, eller kanske du använder dig av din nya tablett till vilken du har laddat ner detta projekt. Kanske du läser arbetet från din hem dator, vart du just installerat en ny jätteskärm, eller kanske du läser i spårvagnen på väg till din arbetsplats från din nya mobiltelefon med en liten skärm. Det viktiga är att förstå att i dagens samhälle utvecklas tekniken med otrolig fart, och trenderna byter ofta. Hur kan vi då implementera nya webbprojekt i en miljö som inte längre håller samma standarder som vara vanliga för bara några år sedan? Före revolutionen bland mobila apparater började kunde webbutvecklarna nästa till 100 % lita på att slutanvändarna använde sig av stationära datorer med oftast nästan lika stora skärmar.

Detta arbete är en undersökning i hur man kan lättast och mest effektivt implementera och förverkliga utveckling av ett responsivt sätt att bygga en webbplats. Då vi använder oss i dagens läge allt mera av mobila verktyg, som smarttelefoner och tabletter kommer behovet av responsiv webbutveckling att öka. Gamla webbplatser kommer inte att fungera som de ursprungligen var menade i de mobila verktyg som kommer till marknaden.

Själva konceptet med mobila webbsidor är inte nytt, men då man ser bak i tiden är en stor del av de mobila sidorna inte gjorda på ett sätt som skulle vara smartast och mest praktiskt. Vad händer då en mobilanvändare vill dela en länk på twitter eller facebook till sin vän som har en bärbar dator med stor skärm, då sidan bakom länken är endast utvecklad för mobilen? Samma gäller om mobilanvändaren öppnar sin e-post och märker att hon fått ett meddelande av sin vän, och meddelandet innehåller en länk till en intressant artikel, men då hon öppnar länken med sin mobil visar det sig att sidan är för

tung och för stor för mobilen. Hur kan detta och liknande problem lösas mest effektivt och med tanke på framtiden och de standarder webben har idag?

1 INLEDNING

Idag använder vi nästan dagligen en informationskälla eller något annat medel som baserar sig på Web 2.0 tekniken. Traditionellt har man använt sig av Web 2.0 tekniken för att söka information från en webbplats, ofta görs informationssökningen med en stationär dator. Som exempel kan vi nämna att t.ex. Facebook, Flickr och Wikipedia använder Web 2.0 tekniken. Då när man introducerar mobiltelefoner eller smarttelefoner uppstår problemet hur applikationen fungerar då slutanvändaren har en liten, ofta under 4 tumskärm.

(History of Web 2.0, 2011)

(What is Web 2.0, 2011)

För bara några år sedan kunde man inte tänka sig att den nya tekniken, som introducerades då smarttelefoner kom på marknaden, skulle revolutionera hela webben. Det var inte många år sedan då man endast tänkte på slutanvändare som använde en stationär dator, ofta med stor skärm. Skärmstorleken växte år efter år, vilket har lett till att flera webbsidor inte fungerar korrekt då slutanvändaren istället för en stationär dator använder en smarttelefon. Vissa webbplatser kan vara utvecklade för skärmar som är stora, man kan hänvisa denna trend till perioden då marknaden såg nya stora LCD skärmar för ett relativt billigt pris. Då kunde man göra design som var normalt mycket bredare än 960 pixlar. En design med detta koncept som har t.ex. över 1220 pixels bredd kommer att vara oanvändbar på en mobilapparat med långsam nätverksuppkoppling.

Försäljningen av mobiltelefoner runt världen har ökat för varje år, och enligt Gartner såldes det år 2010 1,6 miljarder mobiltelefoner, vilket var en ökning på 31,8 procent från år 2009. Försäljningen av smarttelefoner ökade år 2010 med 72,1 procent från år

2009, vilket ledde till att smarttelefoner hade en marknad på 19 procent av alla mobila kommunikationsmedel. Andelen kommer att öka även i fortsättningen.

(Gartner 2011)

I detta arbete kommer jag att undersöka och implementera olika sätt att få en webbplats optimerad för smarttelefoner, och andra mobila verktyg, som surfplattor. Jag kommer att gå igenom olika tekniker hur man kan producera en responsiv webbplats, samt undersöka för- och nackdelarna i de olika teknikerna. Som bas kommer jag att använda mig av Drupal, PHP, HTML5 samt CSS3.

2 WEBBENS UTVECKLING

Sedan starten av Internet boomen, i mitten av 1990 talet har majoriteten av Internet användarna bott i rikare länder, ofta i västvärlden. Förr var datorer mycket dyra, och krävde ofta en relativt bra insikt i hur de fungerade för att de kunde användas. Dessutom krävdes ofta en dyr (och ofta mycket långsam) nätverksuppkoppling, vilken i sig krävde en stark infrastruktur. Denna infrastruktur fanns ofta inte i utvecklings länder.

Tiden har ändrat sig, datorer idag är ofta mycket lättanvändbara och kräver inte samma tekniska kunnande som tidigare. Under åren har priset för hård- och mjukvara sjunkit och man kan idag köpa en dator med Internetuppkoppling relativt billigt, vilket har lett till Internets s.k. globalisering, och idag finns det användare från alla olika världsdelar bortsett från ålder, utbildning eller sociekonomiska klasser.

Webben har vuxit med 480 % sedan år 2000, från 361 miljoner användare (december 2000) till 2095 miljoner, (mars 2011) vilket är över 30 % av världens totalbefolkning. Antalet Internet användare ökade mest i Afrika (2527 %), Mellanöstern (1987 %), Latinamerika (1037 %) och Asien (706 %) (se tabell 1.)

Tabell 1. Tabell över internetanvändare i olika världsdelar. Tabellen innehåller även ökningen under senaste decenniet. (Internet World Stats, 2011)

WORLD INTERNET USAGE AND POPULATION STATISTICS March 31, 2011						
World Regions	Population (2011 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2011	Users % of Table
Africa	1,037,524,058	4,514,400	118,609,620	11.4 %	2,527.4 %	5.7 %
Asia	3,879,740,877	114,304,000	922,329,554	23.8 %	706.9 %	44.0 %
Europe	816,426,346	105,096,093	476,213,935	58.3 %	353.1 %	22.7 %
Middle East	216,258,843	3,284,800	68,553,666	31.7 %	1,987.0 %	3.3 %
North America	347,394,870	108,096,800	272,066,000	78.3 %	151.7 %	13.0 %
Latin America / Carib.	597,283,165	18,068,919	215,939,400	36.2 %	1,037.4 %	10.3 %
Oceania / Australia	35,426,995	7,620,480	21,293,830	60.1 %	179.4 %	1.0 %
WORLD TOTAL	6,930,055,154	360,985,492	2,095,006,005	30.2 %	480.4 %	100.0 %

Då man tar hänsyn till ökningen av internetanvändarna på olika geografiska ställen kan man lägga märke till att internetanvändarna av totalbefolkningen i Västvärlden (Europa 58 %, Nord Amerika 78 % och Australien 60 %) är i medeltal ungefär 65 % av totalbefolkningen. Då man jämför användarna i Västvärlden mot den stigande användargruppen i Afrika (11 %), Mellanöstern (31 %), Asien (23 %) och Latinamerika (36 %) kommer samma procent att endast bli 25 %.

Om utvecklingen fortsätter på samma sätt kommer snart länder och världsdelar som för var en minoritet bland internetanvändare att gå förbi Västvärlden. Då måste man ta hänsyn till olika aspekter gällande hur internet skall fungera. Det kommer att användas andra språk än Engelska. (som idag kan ses som det mest använda språket bland olika mjukvaror och webbplatser) Dessutom kan hårdvaran, nätverksuppkopplingen och infrastrukturen vara helt annan än i Västvärlden. Ifall man vill att alla de nya potentiella användarna skall kunna använda sig av sin webbplats, är det viktigt att den optimeras för alla de olika saker som kan förekomma. (Todd Parker, Patty Toland, Scott Jehl, Maggie Costello Wachs - Progressive Enhancement, 2010)

2.1 Syfte

Syftet med detta arbete är att undersöka med vilka olika tekniker man kan producera en responsiv webbplats. Webbsidan kommer att basera sig på Drupal, vilket är ett mycket

populärt program som kan användas som bas bakom den applikation eller webbplats man utvecklar. Drupal kan användas som ett CMS (Content Management System), ett system för hantering av innehåll, jämfört med andra CMS system är Drupal mycket modulärt och lätt att förlänga med moduler eller andra komponenter. Man kan anse att Drupal är ett CMF (Content Management Framework), ett ramverk för hantering av innehåll, och inte bara en vanlig CMS vilket gör Drupal populärt bland webbutvecklare som använder programmeringsspråket PHP.

Syftet är att forska i olika sätt att producera responsiva webbplatser enligt dagens standarder. Implementeringen kommer att ske med Drupal och ett ramverk som är speciellt utvecklat på tanke på responsiva webbplatser. Under arbetet kommer jag att sträva till att använda bästa praxis inom temat. Målsättningen kommer att vara att undersöka hur man kan utveckla en webbplats som uppfyller alla de krav en responsiv webbplats har. Dessutom skall den fungera på en så stor del av mobila verktyg som bara möjligt och vara framtidssäkrad (future proof) för närframtiden. Då de nya teknikerna (CSS3, HTML5) inte ännu är stödda av alla webbläsare, kommer i själva verket slutprodukten (en responsiv webbplats) att fungera bättre i framtiden då tekniken blir stödd av flera webbläsare, och användare uppdaterar till nya versioner.

I detta arbete undersöks vissa nyckelelement med vilka man kan implementera och utveckla med bästa möjliga redskap en responsiv webbplats eller applikation. Webbplatsen bör funktionera på så många olika medium som bara möjligt och vara lätt att använda, vilket innebär att det bästa sättet är att använda sig av ett existerande system. I detta arbete har Drupal valts som utvecklingsmiljö.

Viktiga nyckelelement jag strävar till att lösa

- Eliminering av gamla tekniken för att utveckla mobilvänliga nätsidor. Den gamla metoden innebär skilda nätsidor för mobilen, ofta i en underkatalog eller ett underdomän. (mobil.domain.com eller domain.com/mobil)
- För att underlätta webbutvecklingen och designen bör det finnas ett stabilt CSS ramverk med flexibilitet.
- Implementation med Drupal

- Användning av nya tekniker, som CSS3 och HTML5
- Flexibel media (bilder, videomaterial mm.)

2.2 Metoder

I arbetet används huvudsakligen experiment vad beträffar olika programvaror och tilläggskomponenter, samt användning av tekniker som jQuery, PHP, JavaScript, HTML5 och CSS3. Tester genomförs med olika program och det granskas hur de fungerar på olika plattformar och hur de beter sig med olika hårdvaror. Till slut genomförs intervjuer med slutanvändarna där meningen är att slutanvändarna skall bedöma om applikationen är lättanvändbar och funktionerar som planerat. Stor vikt kommer att läggas på användning av webbplatsen med en smarttelefon. Programmen som utnyttjas har s.k. öppen källkod, vilket innebär att de är tillgängliga till alla som har intresse att utveckla sin applikation med dem, dessutom kan programmets källkod modifieras enligt behov.

2.3 Avgränsning

Detta arbete kommer att begränsas till att implementera responsiva egenskaper till Drupal samt möjligen göra tester med andra ramverk. Orsaken varför detta system valts är att det är mycket flexibelt och stabilt. Systemet använder sig av LAMP (Linux, Apache, MySQL, PHP) webbplattformen, vilket innebär att detta arbete avgränsas även enbart till denna plattform. Programmen finns även tillgängliga till andra webbservrar och operativsystem, som Windows baserade IIS webbservern, och Microsofts egna MSSQL databas. Alternativt fungerar Drupal även med SQLite databas. Teknikerna kan relativt lätt implementeras till andra CMS system, men detta arbete fokuseras på Drupal arkitekturen. Alla tekniker finns naturligtvis tillförfogande för helt skräddarsydda webbplatser utan ett CMS system.

2.4 Terminologi och förkortningar

Smarttelefon – En mobiltelefon, ofta kapabel till samma funktioner som en stationär dator. Smarttelefoner är mycket populära i dagens läge, de är även kända som ”Smartphones” i detta arbete hänvisar jag ändå till ”smarttelefoner” (Svenska språkets institut får spårk och folkminne 2011, <http://www.sprakradet.se/8327>)

Responsiv – Med detta koncept avser man i webbutveckling ofta att slutprodukten reagerar enligt ett visst argument. I detta arbete kommer argumentet att vara slutanvändarens skärm- eller monitorstorlek.

CMS – Content management system, ett program vilket gör det lätt att behandla information. Informationen är ofta olika PHP variabler, text i databaser samt statiska filer, som t.ex. bilder eller videomaterial.

CMF – Content management framework, ett program som inte enbart är avsett för att bygga traditionella websidor, utan möjliggör även produktion av nätbaserade applikationer, ofta skräddarsydda enligt behov.

Web 2.0 – En webbplats som uppfyller vissa krav, vanlig i dagens tekniker bland webbutveckling.

Drupal – Ett CMS/CMF system vilket möjliggör produktion av webbapplikationer samt webbplatser. Mycket flexibelt och modulärt.

LAMP – En akronym för ”Linux”, ”Apache”, ”MySQL” och ”PHP”. Mycket populär webbplattform som en stor del av applikationer fungerar på. Kan basera sig på ”Perl” eller ”Python” ifall man vill använda ett annat programmeringsspråk än PHP.

Media Query – En CSS3 teknik vilket möjliggör att man kan göra vissa kontextuella förfrågningar till användarens webbläsare. Detta möjliggör att man kan visa information enligt den typen, eller t.ex. bredden som är tillgänglig.

Grid – En mycket använd metod vilket möjliggör snabb utveckling av organiserad HTML kod. Metoden baserar sig på ett CSS rutnät.

Prefix- och suffix – CSS klasser i CSS Gridbaserade ramverk. Klassen används till att skapa mellanrum mellan olika element.

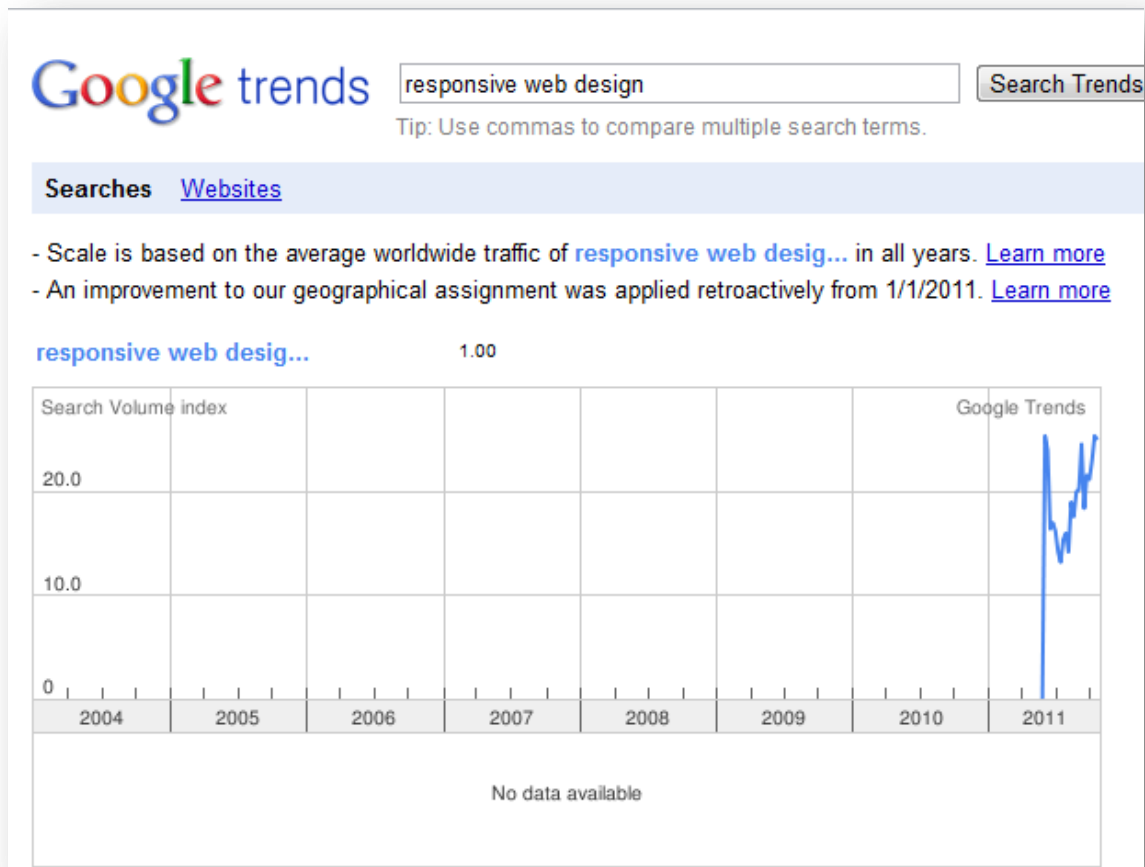
Push- och pull – CSS klasser i CSS Gridbaserade ramverk. Klasserna används till att flytta på element horisontellt.

Gutter – En viss bredd (i pixlar) som används i Grid baserade CSS ramverk. Denna bredd är den bredden som implementeras mellan två kolumner i systemet.

Fluid Grid – Ett sätt att implementera en responsiv layout, till skillnaden från ett vanligt Grid system kommer hela skärmen vara i användning.

3 RESPONSIV WEBBUTVECKLING

Med responsiv webbutveckling menas att slutprodukten skall vara anpassad till olika storlekar av datorskärmar. Konceptet är inte nytt, men håller på att bli en mycket viktig del av webbutveckling idag. Försäljning och användning av mobila verktyg för att använda nätet ökar konstant, och med alla de nya produkter som lanseras årligen syns inte heller något slut till denna trend.



Figur 1. Google trends, sökning med orden "Responsive web design" För all tid det finns data på i Googles databas. Bildkapning tagits 2.11.2011.

Då man gör en sökning på Google märker man att ämnet är relativt nytt, och har ökat radikalt med en början senare hälften av år 2011. (Google Trends 2012) Denna trend kan härledas till att konsumenten idag allt mera vill ha en mobil som fungerar enligt samma principer som stationära datorn. Hit kan räknas alla de nyaste smarttelefoner, surfplattor och liknande verktyg. Tekniken utvecklas med enorm fart, men av alla de webbapplikationer som finns på nätet har bara en bråkdel optimerats så att de fungerar även på mobila verktyg, och av de mobila webbsidor som finns är en stor del gjorts med en metod som inte mera idag anses vara den bästa möjliga. Mobilsidorna är ofta helt skilda sidor som ofta fungerar som ett underdomän (t.ex. mobi.domännamn.com), eller som en skild sida (domännamn.com/mobil) denna typs delning var mycket populär förr, med idag lutar sig trenden samt tekniken mera mot responsiva webbplatser utan skilda mobila undersidor.

Det säljs redan år 2011 fler smarttelefoner än vad det säljs persondatorer vilket tyder på att trenden inte kommer att sluta, utan att bara öka.

(More Smartphones than Desktop PCs 2011)



Figur 2. Google trends, sökning med orden "Responsive web design" under de senaste 12 månaderna. Bildkapning tagits 12.12.2011.

Priset på smarttelefoner kommer att sjunka då de stora företagen måste tävla om kunderna, även de äldre modellernas priser sjunker då nya apparater kommer till marknaden. Vissa företag erbjuder mycket bra och funktionsdugliga smarttelefoner för prisklassen 100 €.

I framtiden kommer det att finnas en stor marknad i utvecklingsländer, där det finns miljoner människor som kanske inte har råd med en persondator, och där det kanske inte finns bredband till förfogande. Istället kan man relativt ekonomiskt bygga ett 3G eller 4G nätverk, vilket leder till att slutanvändaren kan utnyttja sin smarttelefon på bästa sätt.

(Todd Parker, Patty Toland, Scott Jehl, Maggie Costello Wachs. Designing with progressive enhancement, 2010)

3.1 Tre viktiga elementen i responsiv webbutveckling

Termen responsiv webbutveckling innehåller vissa element vilka möjliggör att webbsidan blir responsiv och fungerar på alla storleks skärmar. Att utveckla webbplatser med denna metod ger en stabil och fungerade bas till vilken man kan lätt implementera sin egen design och applikation.

3.2 En flexibel, grid-baserad layout

CSS grid-baserade ramverk finns i otaliga mängder på nätet, alla utvecklare har sitt eget sätt att göra sin eget ramverk, men då man inspekterar ramverken från en högre nivå kan vi anse att de alla finns för att uppfylla samma krav; att få en snygg design utan att hamna programmera samma saker till varje projekt man börjar med.

Ett grid-baserat CSS ramverk är inte responsivt från början, utan de responsiva egenskaperna måste tilläggas skilt. Responsiva egenskaper kan relativt lätt implementeras med CSS3 tekniker, som media queries.

3.3 Media queries

Media queries introducerades vid CSS3, media queries är relativt enkla kontextuella förfrågningar som applikationen och användarens webbläsare utbyter. Iden med media queries är inte ny, utan den fanns redan i CSS2. Vid CSS2 hette egenskapen ”media types”, man kunde med göra små kontextuella krav på hur sidan skulle visas.

(Mediaqueries W3, 2011)

Media typer har funnits redan i CSS2 och HTML4, men i och med CSS3 introducerades ett mera avancerat sätt att använda denna funktion. Man kan tänka sig att en media query är som en fråga till användaren, och någonting ändras (eller ändras inte) då svaret kommer från användaren.

Denna teknik används speciellt vid responsiv webbutveckling, vilket möjliggör att man kan ändra på olika saker då vissa krav möts, dessa krav är oftast krav på användarens skärmstorlek. Media queries begränsas dock inte bara till denna funktion, man kan t.ex. göra en förfrågning om hur användaren har sin surfplatta i bruk, är den i porträttformat eller horisontellt. Förfrågningen lyder som följande:

```
@media all and (orientation:portrait)
```

```
@media all and (orientation:landscape)
```

Dessa olika tekniker ger utvecklaren mycket frihet i hur själva webbprojektet skall utvecklas. Vi kan ha en bättre kontroll över hur information visas till användaren, och att själva applikationen har stöd från de olika webbläsare som finns på marknaden idag.

I detta projekt användes flera olika media queries. Följande media queries (närmar i nästa kapitel) är optimerade för de mobila apparater som är vanligast idag. Ifall det kommer nya trender i framtiden kan man lätt ändra på bredden i de olika satserna, eller skapa helt nya vid behov.

3.4 Media Query syntax

Då CSS3 introducerades innehöll den nya sätt för webbutvecklare att göra sina projekt på syn på slutanvändarna. Ett viktigt element som kom var de nya media queries, vilken gör det relativt lätt för utvecklaren att få webbplatsen optimerad enligt användarens verktyg. Idag stöder en stor del av webbläddrarna media queries, men det finns vissa brister ännu idag. Bristerna hör till de gamla webbläddrarna, och då användare konstant uppdaterar sina webbläsare kommer stödet att öka årvis.

Tabell 2. Tabell över vilka webbläsare stöder media queries.

# CSS3 Media Queries - Candidate Recommendation									
Method of applying styles based on media information. Includes things like page and device dimensions									
Resources: IE demo page with information Media Queries tutorial									
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
3 versions back	6.0	5.0	12.0	3.2	10.6	3.2		10.0	
2 versions back	7.0	6.0	13.0	4.0	11.0	4.0-4.1		11.0	2.1
Previous version	8.0	7.0	14.0	5.0	11.1	4.2-4.3		11.1	2.2
Current		8.0	15.0		11.5		5.0-6.0	11.5	2.3 3.0
Near future	9.0	9.0	16.0	5.1	11.6				4.0
Farther future	10.0	10.0	17.0	6.0	12.0				

Tabell 2. Tabell över vilka webbläsare stöder media queries.

(Stödda webbläsare, 2011)

CSS2 hade även liknande sätt att göra olika frågor till användarens webbläsare, men den tekniken slog aldrig igenom och många webbutvecklare brydde sig inte om att lägga de nödvändiga frågesatserna till webbplatsens kod (Ethan Marcotte 2011, Responsive Web Design) och hela tekniken blev inte använd i den skala som W3C (World Wide Web Consortium, www.w3.org) ursprungligen hade planerat.

Själva media query frågesatsen byggs upp av två komponenter, börjandes med en media typ där man definierar typen på median. Efter denna komponent kommer själva frågesatsen, var man kan definiera det man vill åstadkomma. Man kan tänka sig en media query som ett test till användarens webbläsare.

Som exempel kan vi ta en media query var man frågar efter vilken sorts media typ är tillgänglig och definierar bredden på 1024 pixlar.

@media type (screen) and (min-width: 1024px)

I exemplet ovan är media typen definierad som *screen* och bredden över 1024 pixlar. Ifall användarens webbläsare kommer igenom detta test kommer användaren se det resultat som utvecklaren har bestämt i förhand. Mycket vanligt är att man har olika layouts till olika skärmar, vilket ges ut via media queries, vanligast är ofta media queries

inom 650 pixlar och 1024 pixlar, till denna gräns hör största delen av mobila apparater idag.

Tabell 3. Tabell över media query syntax.

Syntax	Definition	Har min- och max- värde
width	Aktiva områdets bredd (t.ex. bredden på en webbläsare)	Ja
height	Aktiva områdets höjd (t.ex. höjden på en webbläsare)	Ja
device-width	Själva apparatens maximala skärmbredd (t.ex. iPhone4 640 pixlar)	Ja
device-height	Själva apparatens maximala skärmhöjd (t.ex. iPhone4 960 pixlar)	Ja
orientation	Antigen horisontell eller vertikal (syntax: portrait eller landscape)	Ja
aspect-ratio	Aktiva områdets sidförhållande	Ja
device-aspect-ratio	Själva apparatens sidförhållande (t.ex. en iPhone4 3:2, en stationär dator möjligtvis 16:9)	Ja
color	Antalet bitar för färg. (kan vara t.ex. 8-bit eller 16-bit)	Ja
color-index	Antalet färgtoner i själva apparatens färgtabell. (t.ex. min-color-index: 256)	Ja
monochrome	Färgdjupet i en monokrom apparat (bit per pixel, bpp)	Ja
resolution	Densiteten pixlar i apparaten (t.ex. min-resolution: 72 dpi)	Ja
scan	Används då man skärmen är en TV. (värden kan vara progressive eller scan)	Nej
grid	Testar om apparaten är en gammal mobiltelefon med bara en font	Nej

Smala skärmar, tabletter och små bärbara

@media all and (min-width: 740px) and (min-device-width: 740px), (max-device-width: 800px) and (min-width: 740px) and (orientation:landscape)

Satsen ovan används då skärmstorleken är mellan 740 och 800 pixlar bred. Till denna kategori kan tänkas tabletter som används horisontellt eller bärbara datorer med liten skärm.

Normala skärmar, gamla persondatorer eller bärbara datorer

@media all and (min-width: 980px) and (min-device-width: 980px), all and (max-device-width: 1024px) and (min-width: 1024px) and (orientation:landscape)

Satsen ovan används då skärmstorleken är mellan 980 och 1024 pixlar. Denna stil är avsedd för normala persondatorer, vilka har en mindre skärm. Stilen kan även användas då en bärbar dator har en större skärm.

Större vanliga skärmar, eller HD skärmar.

@media all and (min-width: 1220px)

Satsen ovan används då skärmstorleken är större än 1220 pixlar. Den är i stort sett avsedd för vanliga skärmar, men samtidigt för stora HD skärmar.

3.5 Flexibel media

Då man vill att själva webbplatsen är responsiv uppstår vissa problem med bilder och annan media. Webbplatser har mycket ofta bilder, videomaterial eller annat visuellt material som hjälper användaren att hitta den informationen som utsöks. Vissa tjänster är enbart utvecklade för bilder eller videomaterial, som exempel kan jag nämna Youtube.

Vi kan tänka oss att en bild är 800 pixlar bred och 600 pixlar hög. Hur skall vi då kunna implementera bilden till en responsiv webbplats så att bilden fungerar på mobila verktyg där skärmstorleken är bara t.ex. 320 pixlar bred?

Ett alternativ är att man producerar en mindre resolutionsbild vilken fungerar i mobila apparater, men då vi använder oss av denna lösning uppstår ett nytt problem, för då ser bilden mycket liten och suddig ut på stora skärmar.

En lösning skulle kunna dock vara att implementera media queries till olika skärm breddar och höjder och sedan via CSS ge en optimerad bild. Denna lösning skulle kräva flera rader kod, och många olika bilder vilka måste optimeras.

Ett bättre sätt att lösa problemet är att använda flexibla bilder. Denna teknik är relativt simpel, oberoende vad bildens originella storlek är kan vi med CSS och HTML justera hur bilden syns hos användaren. Vi bestämmer bildens bredd och höjd i procent istället än pixlar, vilket leder till då behåller bilden sina ursprungliga dimensioner, även då vi möter en liten skärm.

```
img, video, object, embed {  
    max-width: 100%;  
}
```

Ovan en bit CSS som gör att då vi kallar ”img” klassen med HTML kommer bilderna att vara responsiva. Till CSS har jag för exemplets skull lagt med klasserna video, object och embed, som kan tänkas ha liknande krav vid tanke på responsiv webbutveckling. Vi kan sätta ett videoklipp till vår webbapplikation, och då skärmstorleken minskas skalas videon ner lika som bilder eller annan media.

```

```

Bilderna kan därefter kallas med HTML. Det viktiga att förstå är att bilderna är 100% av den bredden av det element där de ligger. Ifall vi har en bild i ett element som har bredden 400 pixlar kommer bilden naturligtvis att vara 400 pixlar bred. Med detta koncept kan vi utveckla webbplatsen så att bilderna ser bra ut på stora och små skärmar utan att vi tvingas göra olika varianter av bilderna, eller komplicerade javascript, vilka söker reda på användarens monitorstorlek eller operativsystem.

4 TEKNIK

Alla program som används i detta projekt är gratis program, som finns att ladda på internet. Som bas för programmet använder jag Linux distributionen Ubuntu, (www.ubuntu.org) som jag kör i Virtual Box. Orsaken varför jag valt denna miljö är att trots att programmen fungerar på olika plattformar är den mest stabila enligt mina tidigare erfarenheter Linux operativsystemet. Som databas använder jag mig av MySQL, och som programmeringsspråk PHP. Webbservern på Ubuntu är Apache på vilken Drupal fungerar mycket bra. Själva webbplatsen är PHP baserad, men stilerna och strukturen kommer från HTML, JavaScript och CSS.

4.1 Drupal

”Drupal är ett gratis, mycket modulärt innehållshanteringssystem licenserat under GPL (General Public License). Systemet tillåter enskilda individer eller stora grupper av användare att enkelt publicera, hantera och ordna ett brett utbud av innehåll på en webbplats. Till skillnad från andra system är Drupal uppbyggt mer som ett innehållshanteringsramverk bestående av komponenter som kan användas som sådana eller konfigureras helt enligt eget behov. Denna design erbjuder stor flexibilitet samtidigt som den tillåter personer som inte är programmerare att skapa innehållsrika webbplatser. Den här typen av abstraktion och generalisering har en stor fördel jämfört med de mer traditionella uppgiftsorienterade tillvägagångssätten eftersom man kan skapa ny funktionalitet på webbplatsen genom att använda befintliga komponenter i stället för att utveckla helt nya. Denna flexibilitet medför även en viss grad av komplexitet, eftersom man måste förstå hur de olika komponenterna hänger ihop förrän man kan utnyttja dem till fullo.” (Christoffer Lindqvist, 2010)

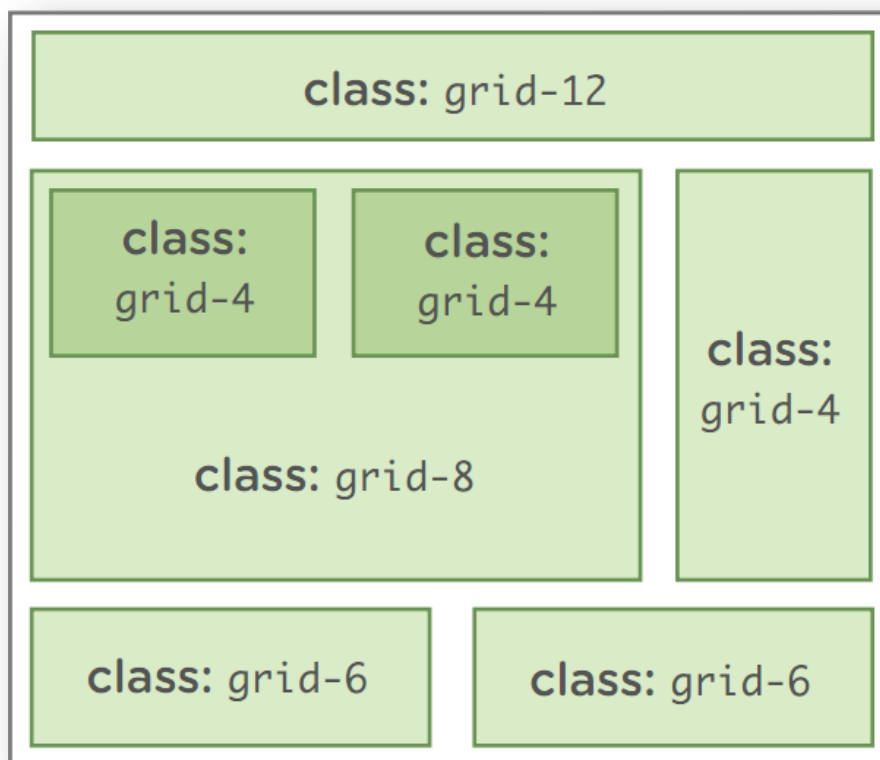
Till den vanliga Drupal installationen har jag dessutom lagt till vissa moduler och ett ramverk som är centralt i detta arbete. Mera om vilka moduler och ramverket i kommande kapitel.

4.2 Grid systemet

Termen Grid härstammar från tiden då man började använda rutnät för att få snygg print, vilket innebar tidningar, böcker och allt emellan. Hela konceptet är gammalt och härstammar från tiden då man började producera print media och ville ha ett välorganiserat och stiligt sätt att visa texten och eventuella bilder. Det finns ingen regel hur själva rutnätet skall se ut, men vanligt är att man har marginaler på alla sidor, storleken kan variera enligt behov. Rutnätet delas ofta i huvudkolumner vilka innehåller text eller bilder. Konceptet är mycket likt Grid baserad webbdesign, och har nästan helt samma principer och tekniker, då ändringar görs i Grid systemet, kommer ändringarna att påverka hur slutanvändaren upplever materialet. (Kata Lyytikäinen, Hannu Riikonen. Painotuotteen Suunnittelu 1995)

Hela konceptet har omvandlats till hur man utvecklar nätsidor och med ett Gridsystem kan man lika som i printmedia visa materialet på ett sätt som är användarvänligt. Tekniken är inte ny, men är mycket populär bland webbutvecklare. Man kan idag tänka sig att man designar för webbläsaren istället än för papper, tidningar, böcker eller annan print. Samma principer gäller fast mediet inte är samma.

Den största skillnaden till Grid baserad design på webben är att Grid konceptet är oftast ett CSS ramverk som är konfigurerat enligt det behov man har. Ett ramverk kan definieras som ”en samling APIs och andra hjälpmedel, vars uppgift är att minska mängden kod och göra utvecklingen lättare och snabbare”. Användning av ramverk har dessutom andra nyttor, man kan alltid använda sig av samma ramverk, och behöver inte göra samma saker flera gånger. Ramverk minskar ofta på olika ”hacker” som ibland måste göras då något inte fungerar rätt (t.ex. Internet Explorer har problem med olika webbt tekniker)

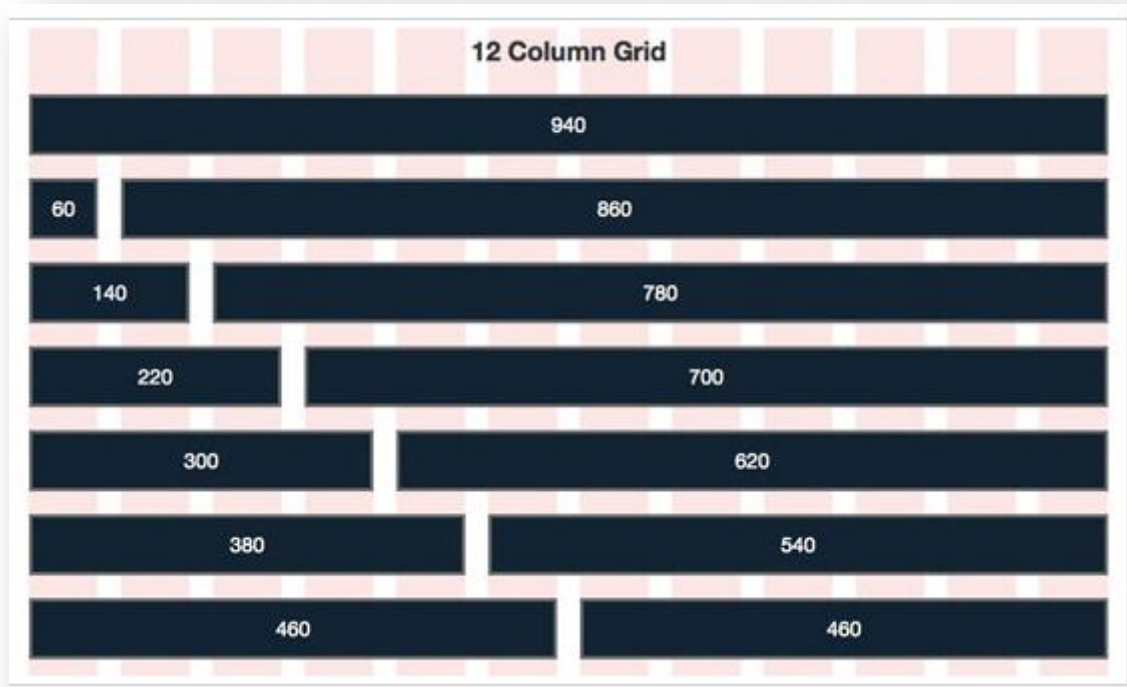


Figur 3. Demonstration av div element i ett Grid system Gridsystem tillåter kapslade (nested) div element, som demonstrerat i figuren.

4.3 Grid baserad webbdesign

Att implementera en Grid baserad struktur till sin webbplats gör det lättare att organisera olika delar av sidan. Då man har ett färdigt s.k. rutsystem (Grid system) vart man lätt kan placera sina element. Detta möjliggör att elementen är horisontellt och vertikalt lika och sidan ser bättre ut. Man kan implementera olika Grid – baserade lösningar enligt behov. Bredden på själva rutsystemet kan man lätt själv bestämma, den bestäms ofta i antalet kolumner, var 12 eller 16 är det mest vanliga. Det finns dock inget som begränsar kolumnernas antal, i vissa komplexa layouts kan det vara nödvändigt att t.ex. implementera en 24 kolumners Grid.

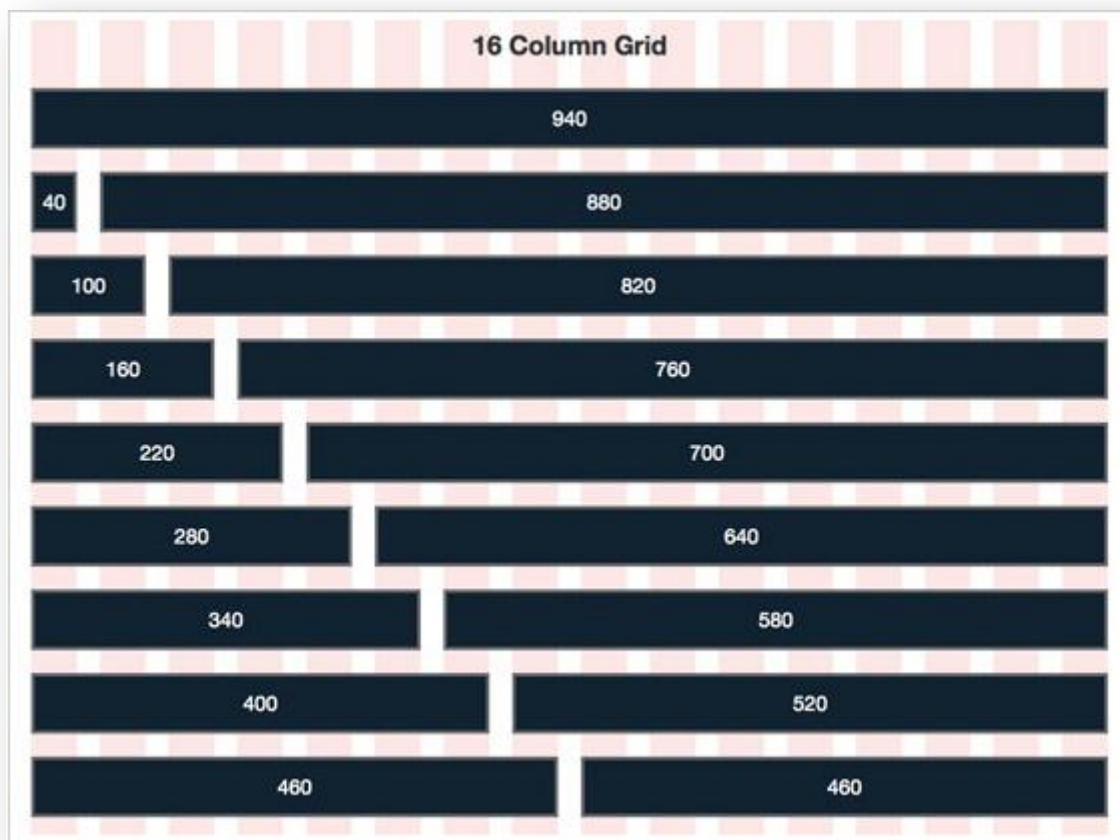
Man delar ofta in kolumnerna enligt bredden på själva layouten. En mycket vanlig lösning är att man gör sitt Grid system 960 pixlar brett, med 10 pixlars kanter (940+10+10=960). Denna lösning ger en 940 pixlar bred användbar yta.



Figur 4. Demonstration på hur en 960 pixlars Grid med 60 pixlars kolumner, 10 pixlar breda marginaler, och 20 pixlar breda "gutters" ser ut.

Ett Grid system väljs ofta som 960 pixlar för att denna siffra går att dela (matematiskt) på många olika sätt, och den passar i dagens läge på flesta monitorer. Då man utvecklar sitt Grid system väljer man hur breda de olika områdena inom Grid systemet är. Kanterna (kallas även gutters) är i detta exempel 10 pixlar, vilket menar 20 pixlar mellan två kolumner. Totalbredden kan då räknas med att räkna kanterna med antalet kolumner i bruk. I figuren ovan kan vi räkna sista raden som följande: $10+460+10+10+460+10$ som svar får vi hela Grid systemets maximala bredd (960 pixlar), vilken är förbestämd.

En annan populär lösning är att istället för 12 kolumner använda sig av 16 kolumner. Vid 16 kolumner kan bredden på Grid systemet vara samma (960 pixlar) men gutter bredden ändras.



Figur 5. Demonstration av ett 16 kolumner Brett Grid system. Till skillnaden från 12 kolumner Brett Grid system är kolumnerna nu 40 pixlar breda.

Orsaken till att man ofta väljer ett visst antal kolumner är för att det matematiskt går att dela kolumnerna på sätt vilket hjälper webbutveckling. Ofta söker man en siffra (antal kolumner) som går att dela jämt med som många tal som bara möjligt. Ifall vi väljer en 12 kolumns Grid kan vi dela de 12 kolumner jämt med två, tre, fyra och sex. Alternativt kan kolumnerna delas in i ojämna tal, som t.ex. fem, fyra och tre.

Denna indelning är mycket viktig att förstå då man bygger på ett Grid system. Idén med att dela in sitt material i kolumner är att man kan organisera materialet enligt behov.



Figur 6. En demonstration över en webbplats med 12 kolumners Grid, samt element med olika CSS klasser i Grid systemet.

Då man delar in materialet man visar i en Grid kan man med olika CSS klasser (ofta färdigt implementerade i Grid ramverket) visa sitt material på det sättet som passar bäst.

I figuren ovan finns tre olika sätt att visa information med CSS klasser i ett Grid system med 12 kolumner. Det översta elementet (menu) har bredden över hela Grid systemet. Inga andra element finns vilket leder till att hela Grid systemets bredd används. CSS klassen (.menu .grid-12) kan inläggas i själva HTML dokumentet.

Det andra elementet har två olika delar, vilka är indelade i Grid systemet så att båda är lika breda. Elementen är delade in med CSS enligt (.header .grid-6) att ge samma breddvärde till båda element.

Det sista elementet har samma funktion som de övriga, till skillnaden är elementet delat in i tre delar (preface .grid-4). Man kan dela in sina element hur man själv ser bäst, de enda krav är att element inte kan vara smalare än en kolumn, eller bredare än maxiantalet kolumner. I exemplet ovan går det inte att dela in element i kolumner som överskrider 12. Man kan dock lätt implementera ett Grid system som är 12, 24, 36 eller bredare. I praktiken finns det inte stor nytta av så breda Grid system, men i teorin är de möjliga.

4.4 Pixel Grid eller Fluid Grid

Webbplatsens utvecklare (eller kunden som beställer ett projekt) väljer ofta i början av projektet hur själva layouten skall implementeras. För vissa ändamål är det bättre att använda ett "Fluid Grid" system. Ofta används dock inte fluid Grid, orsaken är ofta att

webbdesigners inte gillar fluid design. Dessutom fungerar sidan inte enligt designen utan enligt hur sidan används, och detta beror stort på användarens webbläsare.

Detta projekt kommer jag att begränsa till en bestämd bredd (960 pixlar, med 12 kolumner), vilken systematiskt responderar enligt användarens mobila apparat. En annan faktor för att jag väljer förbestämda Grid breddar är att jag personligen har större intresse för dem, det finns mera frihet och utveckla webbplatsen.

(Noora Suurnäkki, Typografia ja gridisuunnittelu verkossa, 2011)

4.5 CSS metoder

Då man arbetar med et Grid baserat system kan man använda sig av olika CSS klasser som ofta hör till Grid ramverk. Klasserna gör det lätt att ändra på olika elements placering på webbplatsen. Man kan välja hur breda elementen skall vara, vilket gör det möjligt att placera element och få funktionalitet från responsiv webbdesign utan att hamna ändra på själva elementen. Då man placerar element i Grid systemet och flytta på elementen, eller ändrar på bredden mellan elementen kan detta utföras med CSS klasserna ”prefix- och suffix” och ”push- och pull”.

(The 960 Grid System made easy, 2012)

En orsak till att man vill använda push och pull CSS klasser är för att göra sökmotor optimeringen så bra som bara möjligt. Materialet kan visas till slutanvändaren på ett sätt, men i själva HTML koden (som sökmotorerna analyserar) kan man med push och pull klasser optimera materialet så att det viktiga kommer först. (Denna teknik kallas ibland även för ”content first”)

(Content First, Luke Wroblowski, 2011)

4.5.1 Push- och pull CSS klasser

Dessa två relativt enkla klasser möjliggör att man kan flytta element i Grid systemet. Ändringen sker visuellt men inte på HTML kod nivå, vilket är effektivt då man tänker på sökmotoroptimering.

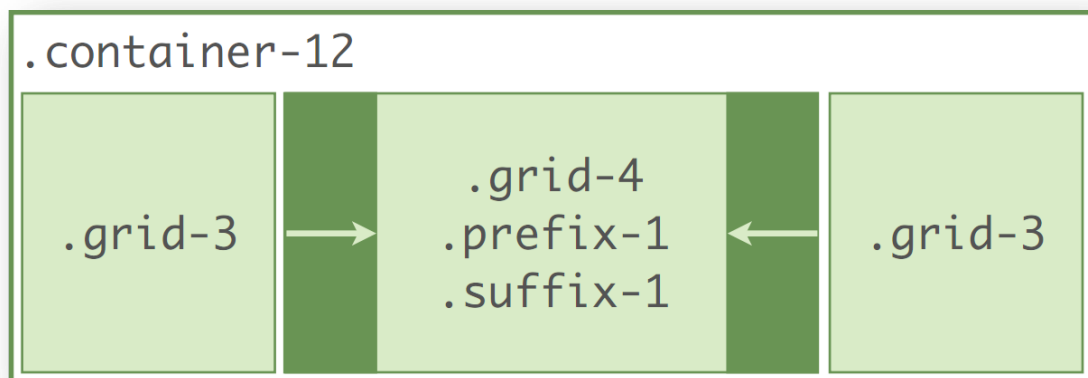


Figur 7. Demonstration hur push och pull metoderna fungerar (i ett 12 kolumners Grid system) i HTML koden, samt visuellt. I figurens övre del finns element utan push och pull klasser, o i nedre delen av figuren kan vi se hur "logo" elementet har bytt plats med "text column" elementet. Push och pull klasserna är insatta i HTML koden enligt figuren.

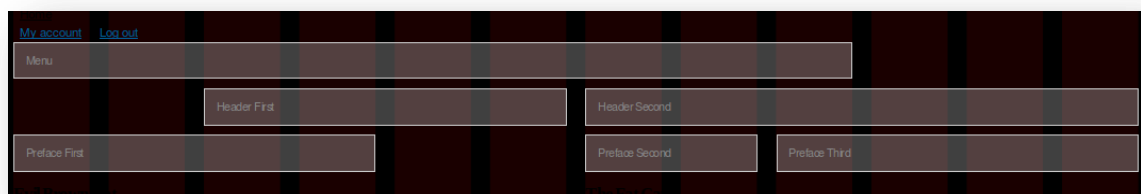
Med CSS klasserna kan vi lätt ändra på olika elements placering utan att byta på ordningen i själva koden, ifall man vill sträva till att ha en viss struktur inom sin källkod är denna teknik mycket användbar.

4.5.2 Prefix- och suffix CSS klasser

Ibland vill man ha tomma element eller platser på sin webbplats, då kan man lätt använda suffix och prefix CSS klasserna för att skapa önskat resultat.



Figur 8. Ett element i ett 12 kolumner brett Grid system med en kolumns prefix och suffix klass. Prefix eller suffix är alltid en kolumn breda. Ovan kan räknas som $3+1+4+1+3$ vilket gör totalt 12 kolumner.



Figur 9. En demonstration över hur prefix och suffix metoder fungerar i Grid systemet. (Jämför med figur 6, vilken inte har prefix eller suffix klasser)

Med CSS klasserna prefix- och suffix kan element flyttas enligt Grid systemet horisontellt. (till höger eller vänster) Figuren ovan innehåller samma element som Figur tre, med några tilläggs prefix- eller suffix CSS klasser. Ifall man ger ett suffix eller prefix värde kommer elementets ursprungliga kolumnbredd att måsta varieras enligt den givna nya klassen. Ifall man vill hålla samma kolumnbredd (totala elementens kolumnbredd överskrider Grid systemets) kommer de element som inte har utrymme på det förbestämda Grid bredden att flyttas till en ny rad. Samma gäller om ett element är placerat på samma rad med andra element och totala bredden överskrider Grid systemets bredd.

Fösta elementet har samma bredd som i figur tre, med ett extra CSS prefix. (menu grid-9 .suffix-3) Man kan märka att elementet har dragits tre kolumner till vänster som resultat av den extra CSS klassen.

Den andra raden innehåller två element, (header-first .grid-6 .prefix-2) men som tilläggs innehåller det första elementet en tilläggs CSS klass (.grid-6 .prefix-2) vilket resulterar i att den har trycks två kolumner till höger.

Den tredje raden innehåller, tre element var endast elementet i mitten har en prefix klass. (preface-second .grid-4 .prefix-2)

(Omega ramverket, Grid systemet 2012)

5 OMEGA RAMVERKET

Omega ramverket (Omega ramverket, 2012) är ett ramverk för Drupal vilket är ursprungligen utvecklad av Jake Strawn. (Jake Strawn, <http://himerus.com/>) Grundprincipen med ramverket är att göra utveckling av responsiva webbplatser lättare, och med rätt standarder. Ramverkets nya version är relativt ny, vilket innebär att all dokumentation inte ännu finns tillgänglig. Ramverket har utvecklats från början med tanke på de

responsiva egenskaper som används idag. Ramverket uppmuntrar webbutvecklare att utveckla sin webbplats först för mobilen, varefter man implementerar den vanliga vyn för bredare skärmar. Omega ramverket tillsammans med Delta modulen (se kapitlet om moduler) kan skapa mycket effektivt olika layout för olika ändamål.

Ramverket erbjuder olika sätt att konfigurera sin webbplats, och kan vid behov använda sig av tilläggsmoduler. De moduler som används regelbundet med Omega ramverket är Delta, Context, Omega Tools och Chaos Tools. Moduler behandlas noggrannare i kapitlet om Drupal moduler.

Ramverkets primära funktioner

- En responsiv Grid layout
- 12, 16 eller 24 kolumners Grid färdigt inbyggt.
- HTML5 eller XHTML (beroende på webbplatsens behov)
- ”Content first” med prefix och suffix CSS klasser
- Aktivera eller inaktivera zoner eller region enligt behov
- Editera olika zoners eller regioners plats med administratörspanelen.
- Insätt egen CSS för önskade visuella ändringar.
- Producera sid-specifik layout med Context modulen.

Ramverket finns på nätet, och kan laddas ner gratis. Ramverket kräver en installation av den nyaste versionen av Drupal 7. Ramverket finns även till Drupal 6, men är inte längre under aktiv utveckling. Ramverket kan även kategoriseras som ett tema till Drupal, men har flera funktioner som gör Omega mera som ett ramverk.

(Omega Framework, 2011)

5.1 Organiserad CSS i Omega ramverket

För att Omega ramverket skall uppfylla kraven för att konstruera responsiva webbplatser har ramverket en speciellt uppbyggd CSS struktur. Man kan editera själv alla CSS filer och så länge man följer de regler som ramverket har behöver man inte ändra på

PHP filerna i Drupal. CSS filerna har en struktur vilket gör det möjligt att gömma vissa element då när användaren har en liten skärm. Detta är effektivt på tanke på att mobilanvändare har ofta en långsammare nätverksuppkoppling än de stationära användarna. I sådana fall kan en bild vara onödig att ladda ner då man har en mobil i användning, det bästa sättet är att man fokuserar på den allra mest relevanta informationen.

Grundprincipen med hur CSS filerna är organiserade är att man skall ha en möjlighet i ett tidigt skede planera och utveckla sin webbplats så att den fungerar i mobila verktyg. Personer eller företag vilka använder sig av ramverket har dock ofta intresse eller som uppgift att göra sin webbplats responsiv, vilket är syftet bakom ramverket. Man kan säga att ramverket är uppbyggt med tankesättet ”mobile first” (utveckling för mobilversionen görs före normala webbplatsen)

5.2 Mobile First

Termen ”Mobile First” kommer från ett idag känt blogg inlägg var Luke Wroblewski skriver om framtidens webbutveckling, och hur man kan med hjälp av en ny filosofi lättare komma fram till resultat då användarna använder mera mobila verktyg att surfa på nätet.

(Mobile First, Luke Wroblewski, 2012)

Ideologin bakom mobile first grundar sig på att webbplatsen byggs upp på tankesättet ”utveckla för mobila verktyg först” med detta tankesätt strävar man efter att utveckla sin webbplats i första hand på tanke på mobilanvändaren. Man kan optimera sidan så att bara den viktigaste och mest relevanta informationen kommer fram då mobilanvändaren öppnar sidan. Därefter kan man börja utveckla webbplatsen till andra målgrupper. Denna ideologi är färdigt inbyggt i Omega ramverket vilket både lättar och gör utveckling snabbare och mer effektiv.

Tekniken bakom detta koncept introducerades med CSS2 och har vidare utvecklats till CSS3 som media queries tekniken i fråga gör det möjligt att göra olika förfrågning till

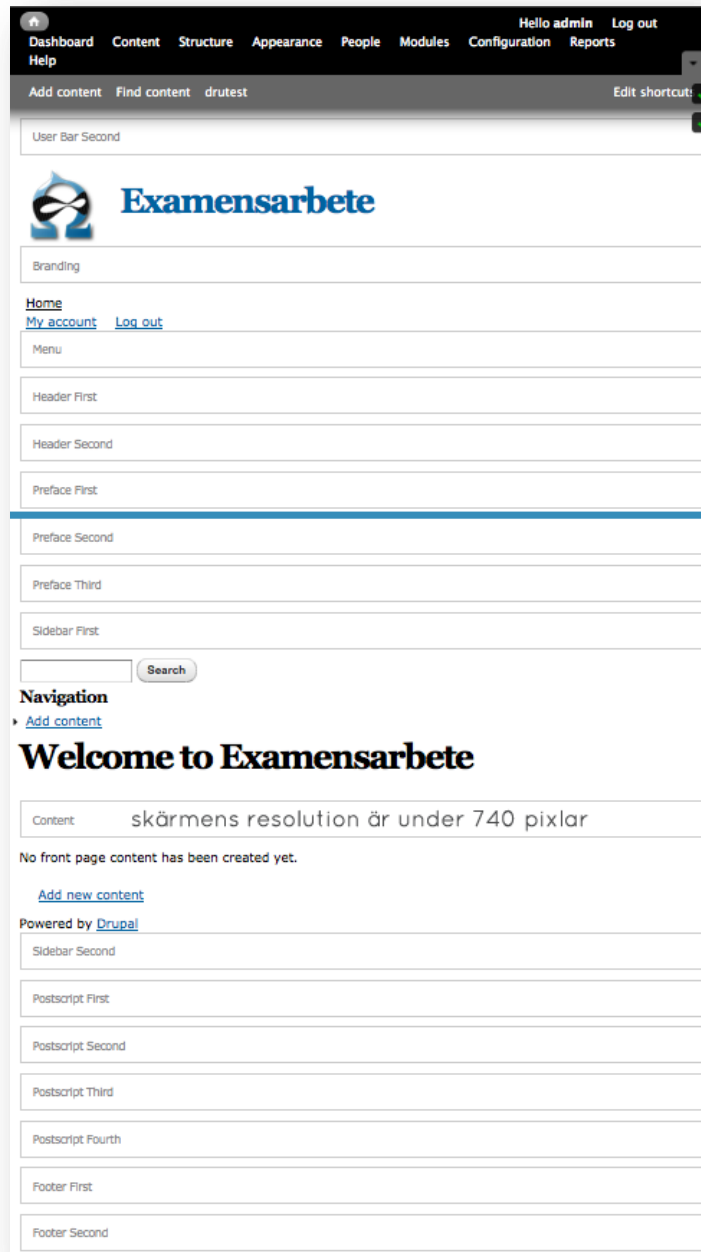
användarens webbläsare och som svar få vilka dimensioner den har. Mera om detta i kapitlet om media queries.

5.3 CSS Strukturen

De olika CSS filerna är indelade in i två huvudgrupper (global.css och alpha-default.css), varav den andra är indelad i en undergrupp med tre CSS filer. Strukturen är baserad sig på att man skall utveckla sin mobila version av webbplatsen först, och först efter det versionen för vanliga persondatorer. De olika CSS filerna tas dynamiskt i bruk, då olika media queries inträffar med användarens webbläsare.

global.css

Denna CSS fil innehåller stilen till hela sidan, och är den viktigaste CSS filen, vilket betyder att andra CSS filer måste överskriva denna fil ifall man vill visa ett visst element med ett visst krav. Till denna CSS fil kommer alla de stilar som webbplatsens mobilversion skall ha.



Figur 10 – Demonstration av resultatet då skärmstorleken möter kraven för media queries. Denna layout är avsedd för mobilen eller små tabletter.

alpha-default.css (delas in i tre underklasser av CSS filer)

Denna CSS fil innehåller stilen till normala persondatorer, som ofta har en större skärm, och resolutionen brukar vara minst 1024 pixlar bred. CSS filen kommer som andra i viktighetsordningen, den skriver över alla andra CSS filer utan global.css. Denna fil har

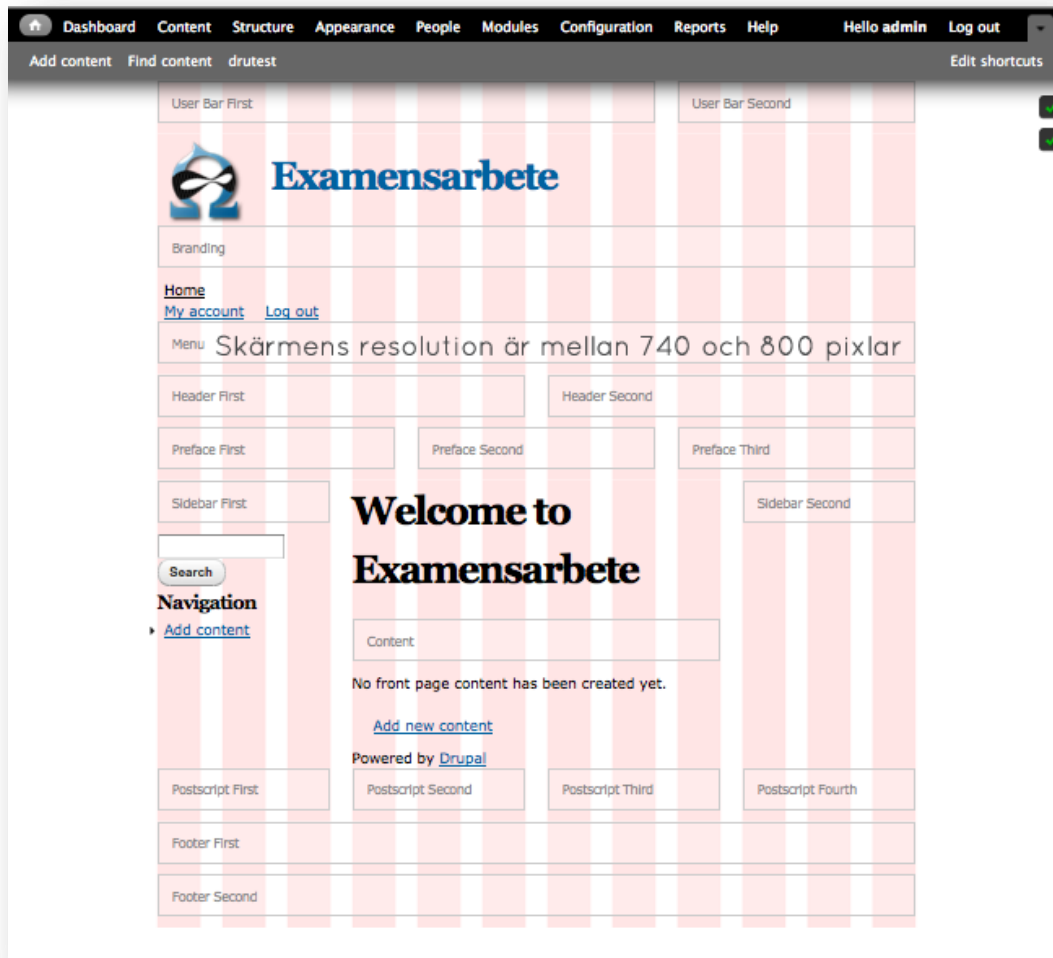
dessutom en specifik roll, enligt CSS har zonerna och regionerna ordnas så att de är relativa till varandra. Med denna egenskap kan alla regioner ordnas på varandra vilket gör det lättare för mobilanvändaren att använda sidan, då man kan lätt hantera sidan med att rulla upp- eller ner med sin mobila apparat. Denna funktion kan implementeras med att modifiera hur Grid systemet visas till slutanvändaren.

```
.grid-1 ... grid-24 {  
position: relative;  
}
```

Med denna teknik kan vi ändra layouten så att alla element (alla Grid klasser) visas till användaren som ett block, och inte bredvid varan, som med de övriga layouter som Omega ramverket implementerar. Då användarens skärm bredd överskrider 740 pixlar implementeras en annan layout. Vi kan då anta att användaren inte längre använder en mobil apparat med en liten skärm.

alpha-default-narrow.css

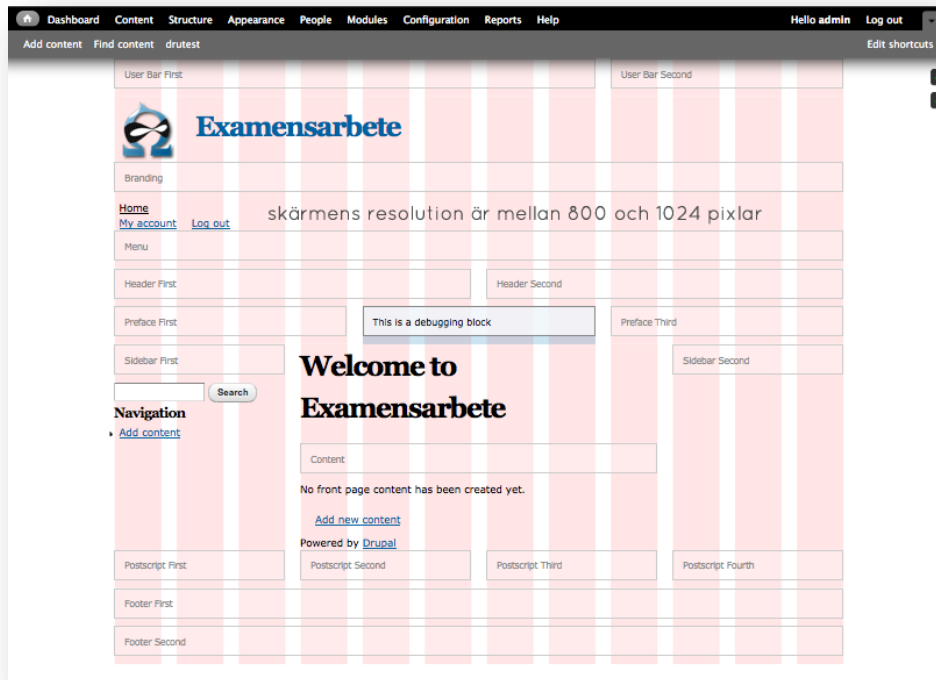
CSS filen har en mera specifik uppgift, denna fil kommer i bruk då användarens webbläsare har bredden mellan 740 och 800 pixlar. Denna bredd är vanlig bland surfplattor och läsplattor, vilket möjliggör att man kan göra en specifik layout för dessa mobila apparater.



Figur 11 – Demonstration av resultatet då skärmstorleken möter kraven för media queries. Denna layout är avsedd i första hand för tabletter.

alpha-default-normal.css

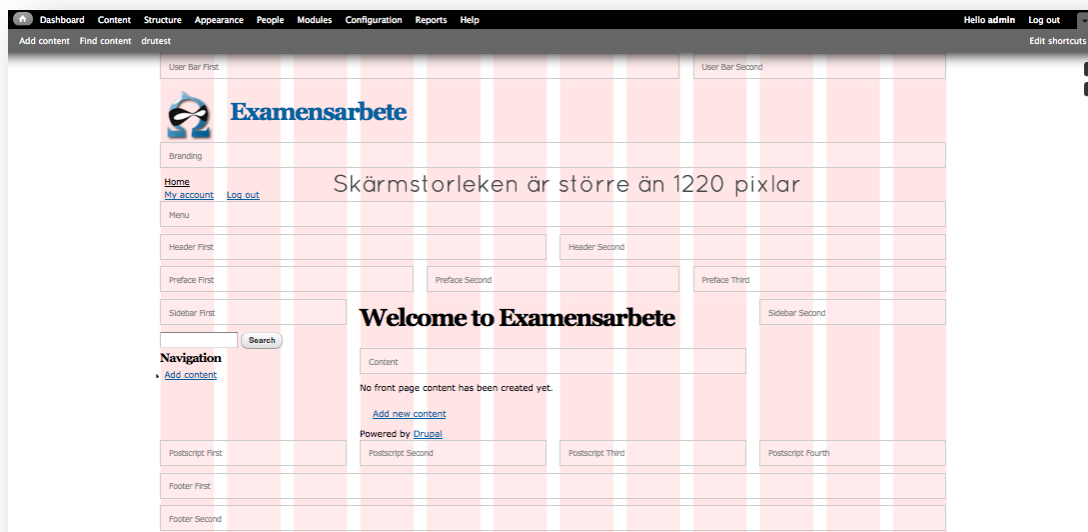
Denna CSS filen tar hand om alla normala monitorer som har bredden mellan 800 och 1024 pixlar. Hit hör ofta mindre monitorer som finns på vanliga persondatorer eller bärbara datorer. Till denna kategori hör även surfplattor som används horisontellt.



Figur 12. Demonstration av resultatet då skärmstorleken möter kraven för media queries. Denna layout är avsedd för bärbara persondatorer eller vanliga datorer med normal skärm.

alpha-default-wide.css

Till den sista CSS filen hör alla monitorer som har stora ofta HD monitorer. CSS filen kommer i bruk då bredden är 1220 pixlar eller större.



Figur 13 – Demonstration av resultatet då skärmstorleken möter kraven för media queries. Denna layout är avsedd för datorer med stor skärm. Denna layout förekommer endast då användarens skärmstorlek överskrider 1220 pixlar.

Denna struktur är inbyggd i ramverket, men alla CSS filer är möjliga och konfigurera själv, man kan i teorin göra oändligt många filer som responsivt tas i bruk då skärmstorleken minskar eller ökar. Ifall man inte vill ändra på stilarna kan man endast sätta upp global.css filen med sin stil, och denna kommer att vara aktiv under hela sidan oberoende av slutanvändarens skärmstorlek. Detta möjliggör att man kan utveckla sin webbplats för en viss storleks skärm, och första senare fortsätta vidare med andra CSS filer.

Optimalt skulle man kunna tänka sig att ramverket fungerar enligt följande:

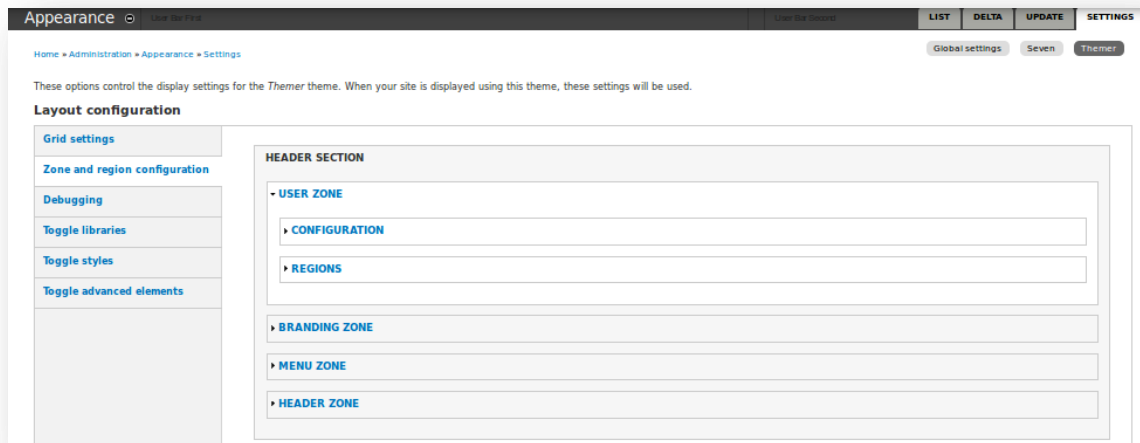
- Skärmen under 740 pixlar = global.css används. Inga andra CSS stilfiler laddas som har direkt påverkan på visuella egenskaper.
- Skärmen mellan 740 och 800 pixlar = alpha-default-narrow.css används, CSS filen skriver över de värden som är samma i global.css
- Skärmen mellan 800 och 1024 pixlar = alpha-default-normal.css används, CSS filen skriver över de värden som är samma i global.css
- Skärmen över 1220 pixlar = alpha-default-wide.css används, CSS filen skriver över de värden som är samma i global.css

- Skärmen den storlek utvecklaren vill = egen-custom.css används (ifall nåt speciellt vill implementeras)

Då ramverket fungerar med att ta i bruk olika CSS filer enligt media queries kan vi gå tillbaka till ”mobile first” ideologin. Nu kan vi lätt se hur lätt det är att först designa för mobila verktyg, och senare till verktyg med större skärm, det ända vi behöver ändra på (om vi så vill) är de värden som kräver modifiering, resten av CSS filerna kan hållas orörda och behöves inte dupliceras.

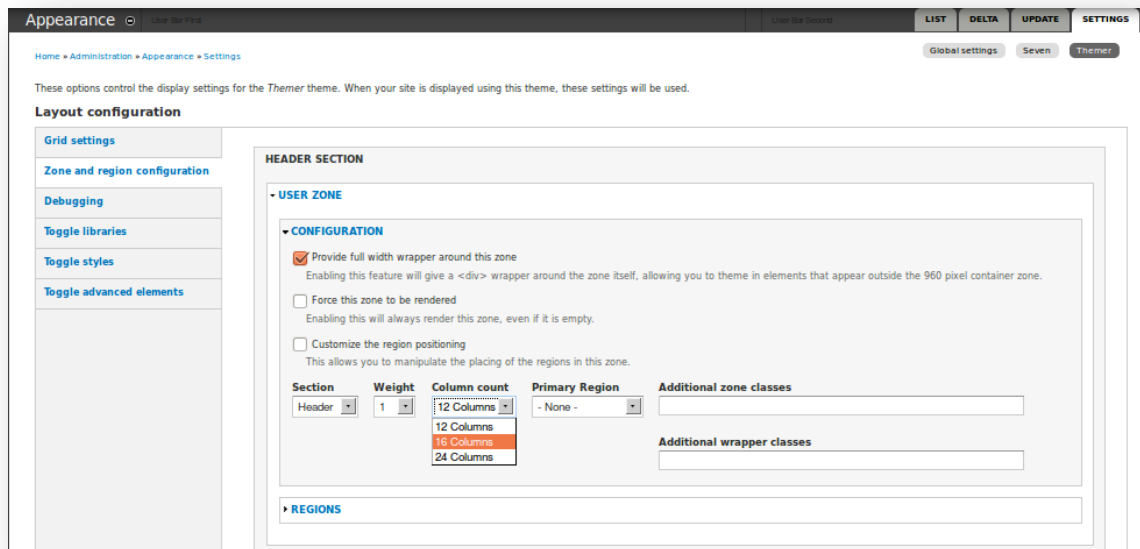
5.4 Omega administration och utveckling

Omega ramverket baserar sig på olika zoner och regioner. Denna egenskap är unik för Omega ramverket, och finns inte i andra Drupal baserade teman. Ramverket bygger på ett 960 pixlar brett (alternativt ett fluid CSS rutnät) CSS rutnät, och då ramverket tar hand om CSS filernas struktur kan man lätt implementera en responsiv applikation. Då skärmstorleken ökar eller minskar betar sig zonerna och regionerna sig enligt CSS filen som används. Som standard läggs alla regioner på varandra då skärmen är så liten att ramverket använder endast de kriterier som kommer från global.css. Men kan såklart anpassa regionerna och zonerna att bete sig som man själv vill. Dessa metoder kan implementeras via CSS. Det viktiga på tanke på sök motor (t.ex. Google, Bing) optimering är att ramverket kan producera ett s.k. ”content first” (Informativa materialet först) sätt att visa material, vilket gör att sökmotorerna indexerar bättre sidan. Till hjälp kan användas olika CSS metoder beskrivna i kapitlet om CSS metoder.



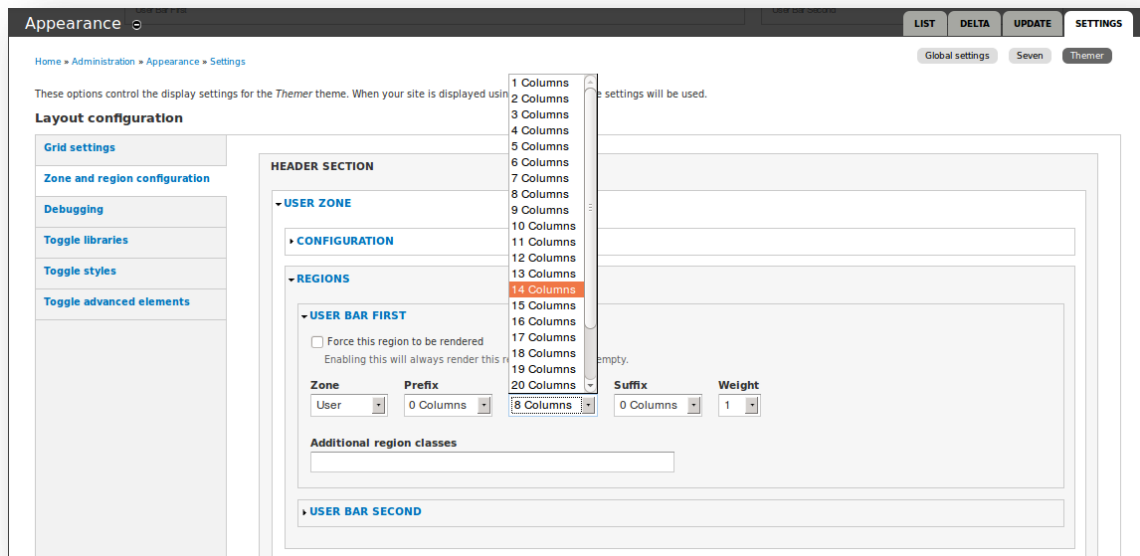
Figur 14. Omega ramverkets administration, var det finns möjlighet att ändra på element i olika regioner. Bilden visar header sektionen, med user, branding, menu och header zonerna.

Omega ramverket låter oss välja bredden på olika zoner med CSS push och pull metoden (mera i kapitlet om CSS push- och pull metoder), vilka renderas till användaren enligt det. Zoner och regioner kan byta plats eller nya kan skapas, vilket kan vara nyttigt då man vill visa en viss layout till användaren, vilket lätt implementeras med Context och Delta modulerna. Ramverket möjliggör alltså att olika layout kan byggas in i samma layout, och använda sig av samma CSS. Som vanligt kräver detta att man gör ändringar med PHP, vilket inte krävs i Omega ramverket.



Figur 15. Header sektionens konfiguration. Ramverket ger möjligheten att ändra på klassens tyngd, i vilken sektion den hör till, hur många kolumner den innehåller samt injektion av tilläggs CSS klasser.

Då vi kan implementera CSS med media queries, och ha vissa kontextuella krav och använda oss av en Grid baserad layout kommer den egentliga kraften fram bakom ramverket. Med media queries kan man bestämma långt hur informationen visas till slutanvändaren. Med små skärmar används global.css filen vilken med Omega ramverket staplar de olika regionerna och zonerna i lodrät form (exempel i figur 3), vilket lättar användningen av webbapplikationen med smarttelefoner eller tabletter med liten skärm.



Figur 16. Konfigurationen för de individuella regionerna, administrations menyn möjliggör positions byte, prefix och suffix, samt bredden av själva regionen.

6 MODULER

Omega ramverket (likasom Drupal) använder sig av olika moduler, vilket gör utvecklingen av projekt lättare. Modulerna är inte obligatoriska men de rekommenderas av Omega ramverkets producerare, och används av en stor del av personer som utvecklar sin webbplats med omega. Dessa moduler är inte enbart menat för Omega ramverket, utan de kan användas i andra applikationer som inte använder sig av Omega ramverket.

6.1 Context

Denna modul möjliggör att man kan ha olika segment av en applikation eller webbplats som aktiveras då vissa krav uppfylls. Kraven kan vara nästan vad som helst, ofta används krav som gäller då en viss meny används, eller en viss nod-typ (node type) aktiveras. Denna modul är mycket populär (context modulen har över 35000 installationer) och kan göra applikationen mera dynamisk.

(Drupal Context Module, 2012)

6.2 Omega Tools

Omega Tools kan användas för att ge Omega ramverket tilläggsfunktioner. Dessa funktioner fungerar även med andra Drupal teman. I detta projekt kommer jag dock inte att gå igenom funktion eller test med andra ramverk eller teman. Omega Tools gör det mycket lätt att exportera konfigurationer som man gjort till Drupal och Omega ramverket. Konfigurationen sparas i Drupals databas och modulen gör det lätt att exportera dem. Omega Tools ger dessutom en möjlighet att återställa temat konfiguration ifall något har gått snett. Omega tools integreras med Drush vilket gör det snabbare att göra ändringar med.

(Drupal Omega Tools, 2012)

6.3 Chaos Tool Suite (Ctools)

Chaos Tools är en modul som inte direkt har en användning. Modulen är en samling nyttiga API:s som andra moduler kan utnyttja. Denna moduls primära funktion är att vara som ett hjälpmedel då man utvecklar moduler. Omega ramverkets vissa egenskaper kräver att denna modul är installerad och aktiverad.

(Drupal Chaos Tool Suite, 2012)

6.4 Delta

Delta modulen var ursprungligen enbart menad för Omega ramverket. Iden med denna modul är att man med hjälp av Context modulen kan göra (i teorin oändligt) olika layouts som baserar sig på den primära layouten utan att göra ändringar i PHP. Modulen fungerar så att den kopierar de inställningarna man har gjort och därefter kan man med Context modulen visa denna kopia. Kopian kan även konfigureras enligt behov. Kopian är ofta olik från primära layouten och visas till användaren då ett kontextuellt krav före-

kommer. Som exempel kan man ha en kopia av sin primära layout och göra önskade ändringarna i regioner och zoner varefter man med Context modulen gör ett villkor och när detta villkor möts visas den nya layouten.

(Drupal Delta Module, 2012)

6.5 Drush

Drush är ett program (Shell script) som används via terminalen (command line interface) och är inte likt de traditionella Drupal modulerna. Drush integreras direkt med operativsystemet och används alltså från operativsystemets terminal. Drush har utvecklats för Linux, Mac och Windows. Drush gör utvecklingen av olika webbprojekt mera effektivt, eftersom man inte behöver använda Drupal's grafiska användarsnitt som kan vara långsamt beroende på datorn man använder. Drush kan (med olika kommandon) dessutom ladda ner moduler, installera moduler och sätta upp nya projekt på bara några sekunder, vilket gör Drush mycket behändigt man håller på att utveckla sin webbplats.

Några exempel med Drush

```
$ drush quickstart-create --domain=virtuell.adress.com
```

Då man skriver in detta skript i terminalen kommer Drush att ladda ner den nyaste versionen av Drupal, installera den med givna användar- och lösenord, installera och konfigurera en databas (mysql) konfigurera Linux DNS server så att man kan med en virtuell adress komma åt sitt nya projekt via webbläsaren.

```
$ drush quickstart-delete --domain=virtuell.adress.com
```

Detta kommando gör det motsatta, och raderar allt som det första skriptet installerade och konfigurerade.

Till Drush kan man göra egna shell skript enligt eget behov. Vardagliga uppgifter som att köra cron (`$ drush cron`), eller att tömma Drupal cache minne (`$ drush cc`) går mycket smidigt med denna modul.

(Drupal Shell (Drush), 2012)

7 AVSLUTNING OCH DISKUSSION

I detta arbete har jag gått igenom olika tekniker som kombinerat ger en mycket bra och stabil grund för att bygga framtida webbplatser. Då man kombinerar olika koncept och implementerar dem till ett ramverk (Omega ramverket) kommer man över vissa problem som kanske skulle normalt kräva extra arbete. Ramverket innehåller en effektiv administration meny med vilken man kan göra ändringar samt ändra på webbplatsens konfiguration. Ramverket är använt av många och under aktiv utveckling.

En annan viktig observation är att man måste se skillnaden mellan en traditionell mobil webbplats, och konceptet med en responsiv webbplats. De traditionella mobila webbsidorna har ofta ett sätt att identifiera användarens mobila verktyg med "user agent sniffing" vilken identifierar med vilket system används med "navigator.userAgent" metoden. Denna metod är mycket osäker, pga. användaren kan själv konfigurera vilken "user agent" som används. Responsiva webbplatser implementerar nya tekniker inom identifiering vilka är mycket säkra och kan inte direkt manipuleras av användaren.

Då man kombinerar ett responsivt Grid ramverk, flexibel media, och media queries kan man utveckla webbplatser för de konstant ökande mobila användare, och samtidigt ha en webbplats som utan skilda mobila undersidor (t.ex. mobi.domän.com) fungerar på alla storleks monitorer, smarttelefoner och tabletter.

8 RESULTAT

Då man använder Omega ramverket och metoderna beskriva i detta arbete kan man relativt lätt producera en webbplats med responsiva egenskaper. Ramverkets nyckelfunktion är användning av media queries, vilket oundvikligt leder till att denna egenskap inte fungerar på gamla Internet Explorer (fungerar inte på äldre IE versioner än IE9, vilken stöder media queries) webbläsare. Detta problem är dock inte kritiskt för att denna bläddrare inte finns tillgänglig till smarttelefoner eller tabletter, vilket betyder att den

responsiva egenskap som media queries ger inte påverkas av detta problem. Slutliga sidan fungerar normalt i Internet Explorer då man har en vanlig persondator i användning.

Som slut till detta arbete visade jag olika webbplatser med mobila verktyg, samt persondatorer, till en liten testgrupp. Enligt testgruppen fungerade alla webbplatser med responsiva egenskaper bättre på mobila verktyg, samt surfplattor. Den största skillnaden var dock användning av smarttelefoner, var skillnaden kom allra mest fram. Minst synes skillnad på persondatorer med stor skärm, vilket var väntat. Testgruppen använde sig av mobila verktyg som; HTC Desire HD, iPhone4, iPad och en persondator med en 22 tumms skärm. (HP w2207)

KÄLLOR

Worldwide mobile sales | 2011, [www], Hämtat 14.09.2011
Hämtat 12.12.2011 <http://www.gartner.com/it/page.jsp?id=1543014>

History of Web 2.0 | 2011, [www], Hämtat 14.09.2011
Hämtat 10.12.2011 [http://aspalliance.com/1135 Myths and Realities of Web 2.0](http://aspalliance.com/1135_Myths_and_Realities_of_Web_2.0)

What is Web 2.0 | 2011, [www], Hämtat 14.09.2011
Hämtat 10.12.2011 <http://oreilly.com/web2/archive/what-is-web-20.html>

The Emca Standard | 2011, [pdf], The Emca standard. Hämtat 10.12.2011
<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

HTML 5 Wikipedia | 2011, [www], Hämtat 11.11.2011
<http://en.wikipedia.org/wiki/Html5>

HTML History w3 | 2011, [www], Hämtat 14.11.2011
<http://dev.w3.org/html5/spec/introduction.html#history-1>

Drupal | 2011, [www], Hämtat 20.12.2011
<http://drupal.org/>

Drupal Wikipedia | 2011, [www], Hämtat 20.12.2011
<http://en.wikipedia.org/wiki/Drupal>

CSS w3 | 2011, [www], Hämtat 20.12.2011
<http://www.w3.org/TR/CSS/>

CSS3 Wikipedia | 2011, [www], Hämtat 19.12.2011
http://en.wikipedia.org/wiki/CSS3#CSS_3

LAMP Wikipedia | 2011, [www], Hämtat 20.11.2011
[http://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](http://en.wikipedia.org/wiki/LAMP_(software_bundle))

Smartphones have conquered PCs | 2011, [www], Hämtat 20.12.2011
http://money.cnn.com/2011/02/09/technology/smartphones_eclipse_pcs/index.htm

More Smartphones than Desktop PCs by 2011 | 2011, [www], Hämtat 11.12.2011
http://pcworld.com/article/171380/more_smartphones_than_desktop_pcs_by_2011.html

API Wikipedia | 2011, [www], Hämtat 12.01.2012
<http://sv.wikipedia.org/wiki/API>

Omega Framework | 2011, [www], Hämtat 12.01.2012
<http://drupal.org/project/omega>

Mediaqueries w3 | 2011, [www], Hämtat 12.01.2012
<http://www.w3.org/TR/css3-mediaqueries/>

Google Trends | 2012, [www], Hämtat 20.01.2012
<http://www.google.ca/intl/en/trends/about.html>

Omega ramverket, Grid systemet | 2012, [www], Hämtat 20.01.2012
<http://omega.developmentgeeks.com/documentation/960grid/push-and-pull-classes>
<http://net.tutsplus.com/tutorials/html-css-techniques/mastering-the-960-grid-system/>

Mobile First | Luke Wroblewski | 2012, [www], Hämtat 20.01.2012
<http://www.lukew.com/ff/entry.asp?933>

Drupal Context Module | 2012, [www], Hämtat 20.01.2012
<http://drupal.org/project/context>

Drupal Omega Tools | 2012, [www], Hämtat 20.01.2012
http://drupal.org/project/omega_tools

Drupal Chaos Tool Suite | 2012, [www], Hämtat 20.01.2012
<http://drupal.org/project/ctools>

Content First, Luke Wroblowski, 2011
<http://www.lukew.com/ff/entry.asp?1372>

Drupal Delta Module | 2012, [www], Hämtat 20.01.2012

<http://drupal.org/project/delta>

Drupal Shell (Drush) | 2012, [www], Hämtat 20.01.2012

<http://drupal.org/project/drush>

Internet World Stats | 2011, [www], Hämtat 10.9.2011

www.internetworldstats.com/stats.html

Stödda webbläsare | 2011, [www], Hämtad 20.11.2011

<http://caniuse.com/#search=media>,

The 960 Grid System made easy, 2012 [www], Hämtad 12.1.2012

http://sixrevisions.com/web_design/the-960-grid-system-made-easy/

Typografia ja gridisuunnittelu verkossa, 2011, Noora Suurnäkki

<http://publications.theseus.fi/handle/10024/27281>

Robert J. Townsend. 2010. Foundation Drupal 7. Friends of Publishing (Apress)

(ISBN: 978-1-4302-2808-0)

Dan Cederholm. 2010. CSS3 for Web Designers. A Book Apart Publishing

(ISBN: 978-0-9844425-2-2)

Lynn Beighley. 2010. jQuery. Wiley Publishing Inc.

(ISBN: 978-0-470-58445-3)

Ethan Marcotte. 2011. Responsive Web Design. A Book Apart Publishing

(ISBN: 978-0-9844425-7-7)

Todd Parker, Patty Toland, Scott Jehl, Maggie Costello Wachs. 2010. Designing with progressive enhancement. New Riders Publishing

(ISBN: 978-321-65888-3)

Kata Lyytikäinen, Hannu Riikonen. 1995. Painotuotteen suunnittelu
(ISBN: 951-719-209-6)

Mobile First, Luke Wroblewski, A Book Apart, 2011
(ISBN: 9781937557027)

