



Webbutveckling med HTML5 Boilerplate

En studie i mallens egenskaper och god utvecklingspraxis

Tom Ahola

Examensarbete
Informations- och medieteknik
2013

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	4190
Författare:	Tom Ahola
Arbetets namn:	Webbutveckling med HTML5 Boilerplate
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	
<p>Sammandrag:</p> <p>I arbetet stiftar vi bekantskap med webbutvecklingsmallen HTML5 Boilerplate och dess innehåll. HTML5 Boilerplate är en populär startbotten för att utveckla snabba, hållbara och anpassningsbara webbsidor och -applikationer. Den innehåller mängder av tekniska lösningar som anses vara bästa praxis när det gäller webbutveckling. Hundratals professionella webbutvecklare har tillsammans bidragit till Boilerplate-projektet och delat med sig av sin kunskap. Strukturen i Boilerplate-paketet behandlas del för del. Paketet är licenserat under en MIT-licens och kan fritt användas för både personligt- och kommersiellt bruk. Flera välkända företag och sidor har valt att använda denna mall. I arbetet undersöker jag vad som gör denna mall kraftfull och försöker identifiera det som är bästa webbutvecklingspraxis. I arbetet blir slutprodukten en responsiv Mobile First Boilerplate-sida. Som utgångspunkt används Initializr, en Boilerplate-mallgenerator som ger oss en anpassad botten. I arbetet läggs stor vikt på mobilvänlig utveckling, något som vi inte kan frånse i dagens läge då mobila apparater används allt mer.</p>	
Nyckelord:	HTML5, Boilerplate, CSS3, jQuery, Initializr, responsiv webbdesign, Mobile First
Sidantal:	34
Språk:	Svenska
Datum för godkännande:	12.6.2013

DEGREE THESIS	
Arcada	
Degree Programme:	Information and Media Technology
Identification number:	4190
Author:	Tom Ahola
Title:	Web development using HTML5 Boilerplate
Supervisor (Arcada):	Johnny Biström
Commissioned by:	
<p>Abstract:</p> <p>In this study we dive into the HTML5 Boilerplate web developing template and its contents. HTML5 Boilerplate is a popular starting ground for the development of fast, durable and adaptable websites and applications. It contains a large amount of technical solutions considered to be best praxis in web development. A community of hundreds of professional developers around the world have contributed to the Boilerplate project and shared their knowledge. This thesis goes through the structural content of the Boilerplate package, part by part. The package is licensed under an MIT-license, free for personal and commercial use. Many well-known companies and sites are already using this template. In the process, I examine what makes this template so powerful and try to identify what exactly are best web development practices. The end product of this work is a responsive Mobile First Boilerplate website. As starting ground, the templates generator Initializr is used, which gives us a customizable Boilerplate template. In this study we put great emphasis on mobile-friendly development, something we can't disregard in today's time when mobile devices are increasingly used.</p>	
Keywords:	HTML5, Boilerplate, CSS3, jQuery, Initializr, Responsive web design, Mobile First
Number of pages:	34
Language:	Swedish
Date of acceptance:	12.6.2013

INNEHÅLL

1	INTRODUKTION	6
1.1	Bakgrund	6
1.2	Syfte	6
1.3	Avgränsning.....	7
1.4	Terminologi.....	8
2	HTML5 BOILERPLATE.....	9
2.1	Beskrivning	9
2.2	Vem använder H5BP?.....	9
2.3	Nedladdning av H5BP-paketet	10
2.3.1	<i>Fulla versionen</i>	10
2.3.2	<i>Anpassade versioner</i>	11
2.4	Innehållet i H5BP.....	11
2.4.1	<i>index.html - märkspråksfilen</i>	11
2.4.2	<i>CSS – stillkonfigureringsfilerna</i>	15
2.4.3	<i>JS - JavaScript- och jQuery-filer</i>	17
2.4.4	<i>.htaccess - Apache HTTP serverkonfigureringsfilen</i>	19
2.4.5	<i>robots.txt – kontroll över webbrobotindexering</i>	19
2.4.6	<i>Övrigt</i>	20
3	SKAPANDET AV EN BOILERPLATE-SIDA	21
3.1	Beskrivning av sidan.....	21
3.1.1	<i>Responsiv webbdesign och Mobile First</i>	21
3.2	Förberedelser och Initializr-botten.....	22
3.2.1	<i>Egen design</i>	25
3.3	Placering av innehåll och kodskrivande	26
3.4	Övriga inställningar.....	27
3.5	Resultat och tester.....	28
4	DISKUSSION	30
	Källor	32

Figurer

Figur 1. HTML5 Boilerplate logga (hämtad från html5boilerplate.com)	9
Figur 2. Exempelstruktur på H5BP-paket (kopia från html5boilerplate.com)	11
Figur 3. Raderna 1-18 i index.html (skärmbild från Notepad++)	12
Figur 4. Raderna 19-40 i index.html (skärmbild från Notepad++).	14
Figur 5. Exempel på oönskad textskugga (hämtad från twitter.com/miketaylr/status/12228805301 13.5.2013).....	15
Figur 6. Funktion i plugin.js som undviker JavaScript-fel i webbläsare utan konsol (skärmbild från Notepad++).	18
Figur 7. Exempelkodsnutt i robots.txt (skärmbild från Notepad++).	20
Figur 8. Steg 1, Vi väljer en responsiv botten på Initializr.com . (skärmbild)	22
Figur 9. Steg 2, där jag ytterligare väljer vilka delar jag vill ha med i botten. (skärmbild från Initializr.com).....	23
Figur 10. Responsiv design på botten från Initializr.com (skärmbilder från webbläsaren).	24
Figur 11. Egen design på modellsidan gjord i Adobe Fireworks.	25
Figur 12. Titel och meta-beskrivning ifylld (skärmbild från Dreamweaver).	26
Figur 13. En Media Query som riktar sig på skärmar bredare än 480px.	26
Figur 14. En enkel ikon för min modellsida.	27
Figur 15. humans.txt ifylld med information om vem som skapat sidan och vilka tekniker som använts.	28
Figur 16. Slutgiltiga modellsidan i webbläsare (skärmbild).	29
Figur 17. Slutgiltiga modellsidan testad med HTC Sensation -mobil (skärmbild).	29
Figur 18. Slutgiltiga modellsidan testad med Iphone 4 (skärmbild).	30

1 INTRODUKTION

1.1 Bakgrund

Nya webbprojekt påbörjas ofta med stor entusiasm och iver. Det dröjer inte länge ändå förrän projektkatalogen börjar kännas stökig med flera underkataloger och filer. Dessutom, hur kan man vara säker på att man som utvecklare får med allt det viktiga? Lanseringen närmar sig och man märker kanske att man saknar meta-data på någon viktig sida, något ser konstigt ut i en äldre webbläsare eller man har helt enkelt glömt, att sidan skall se bra ut också som utskriven. Sedan bör vi ju idag tänka speciellt mycket på de mobila apparaterna. Enligt StatCounter var den mobila internetanvändningen 16,77% av den totala i april 2013 i Finland, och trenden den vet vi, är bara ökande (StatCounter 2013). Hur ska vi webbutvecklare då komma snabbast och säkrast igång?

I detta arbete har jag studerat något som kanske är en godtagbar lösning på problemen presenterade ovan. En mall, eller en botten som kommit till utav frustrationen att alltid behöva börja från noll, osäker och utan vägledning. Denna mall heter HTML5 Boilerplate och är en samling verktyg för att man snabbt skall kunna starta upp sitt webbprojekt enligt bästa utvecklingspraxis. Jag har valt detta ämne, för att det ligger i mitt intresse att kunna skapa så välfungerande och moderna sidor som möjligt. I skrivande stund har jag arbetat några år som webmaster på en resebyrå. Jag introducerades första gången till HTML5 Boilerplate under en mobilwebbskurs jag deltog i via min arbetsplats. Där blev jag imponerad av hur kraftfull denna mall verkade, men samtidigt funderade jag, hur enkel är den att ta i bruk och administrera?

1.2 Syfte

Syftet med detta arbete är att ingående undersöka innehållet i HTML5 Boilerplate-mallen och se vad den som helhet och i detalj erbjuder webbutvecklaren. Arbetet kommer att presentera strukturen i HTML5 Boilerplate-paketet och gå igenom de olika delar man valt att ha med för bästa praxis. Slutligen utnyttjas HTML5 Boilerplate-mallen för ett modellprojekt varefter skribenten reflekterar över sina iakttagelser och erfarenheter.

1.3 Avgränsning

Examensarbetet kommer inte att gå in på djupet i de tekniska lösningar HTML5 Boilerplate presenterar eller hur man kommit fram till att just de följer bästa praxis. I utförande- och exempeldelen kommer inte en fullständig webbplats att presenteras, utan produkten blir en modell, där Boilerplates egenskaper testas. Av läsaren väntas grundkunskaper i modern webbutveckling och olika utvecklingspråk som används i arbetet kommer inte att utbildas. Däremot dyker referenser upp i texten, där den intresserade kan studera ämnen noggrannare. Själva forskningsmaterialet i detta arbete har till stor del bestått av dokumentationen i HTML5 Boilerplate, böckerna HTML5 Boilerplate Web Development av Divya Manian, Responsive Web Design av Ethan Marcotte och Mobile First av Luke Wroblewski. Därtill har ett flertal Internetkällor använts.

1.4 Terminologi

HTML5 – Den moderna webbens kommande standard märkspråk som ännu utvecklas. Har sina rötter i tidigare versioner såsom XHTML och HTML4.

CSS3 – Nyaste versionen av Cascading Style Sheets med många moderna förbättringar som fungerar speciellt bra tillsammans med HTML5.

jQuery – Ett mångsidigt JavaScript-bibliotek med crossbrowserstöd som gör det enkelt att programmera dynamiska sidor eller använda färdiga tredjeparts tilläggsdelar.

H5BP – Förkortning av HTML5 Boilerplate som används flitigt i detta arbete.

Responsiv webbdesign – En term och ett utvecklingsätt av nätsidor som anpassar sig till det media och de skärmstorlekar de presenteras på.

Mobile first – En utvecklingsfilosofi eller ett tillvägagångssätt där man prioriterar mobila apparater och små skärmstorlekar.

IE – Internet Explorer-webbläsare utvecklade av Microsoft.

GCF – Google Chrome Frame, en tilläggsdel till äldre versioner av Internet Explorer som möjliggör presentation av moderna webbfunktioner.

CDN – (Content delivery system) Ett större system av servrar vars uppgift är att leverera innehåll åt slutanvändaren med god prestanda och tillgänglighet.

grid – En term för det mönster och de ramar man använder som grund i sin webbdesign.

2 HTML5 BOILERPLATE

2.1 Beskrivning

HTML5 Boilerplate är en populär startmall för att utveckla snabba, hållbara och anpassningsbara webbsidor eller -applikationer. Det är ett projekt som kommit till under flera år av iterativt utvecklande och är en sammansättning av kunskap från en gemenskap professionella webbutvecklare runtom världen. Mallen binder inte användaren till någon viss utvecklingsfilosofi utan är fritt modifierbar och öppen för vidareutveckling. Paketet är licenserat under en MIT-licens och kan användas gratis både för personligt- och kommersiellt bruk. (HTML5 Boilerplate 2013a)

År 2009 medan Paul Irish ännu arbetade med andra projekt, lade han märke till att det fanns kodsnuttar han regelbundet var tvungen att kopiera och klistra in i sina webbsidor. Han började samla dessa i ett enda dokument och till slut formades det till en mall som även gjordes offentlig. Mallen uppskattades och fick fantastisk respons av andra webbutvecklare som började anpassa och följa riktlinjerna i dokumentet. Projektet HTML5 Boilerplate startade Irish som en följd av detta i januari 2010. Med tiden kom allt fler professionella webbutvecklare att medverka i projektet och bidra till H5BP:s innehåll. Idag har hundratals webbutvecklare bidragit, projektet har utvecklats snabbt och dess popularitet är större än någonsin. (HTML5 Boilerplate 2012)



Figur 1. HTML5 Boilerplate logga (hämtad från html5boilerplate.com)

2.2 Vem använder H5BP?

Några välkända företag som använder H5BP är Google, Microsoft, Unilever och Barack Obamas kandidatsida. På H5BP:s sidor finns en referens till en längre lista över sidor som använder mallen (se *H5BP sites* 2013). Det finns också en sida som fungerar som ett visningsgalleri över H5BP-sidor (se *H5BP.net* 2013). Förutom webbsidor har H5BP också vidareutvecklats till tusentals övriga mallar och CMS- system (se *H5BP forks*

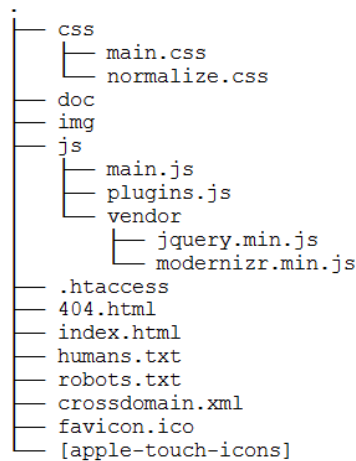
2013). Igenkännbart förutom källkoden är då utvecklarna lämnat kvar något som blivit lite till H5BP:s varumärke, nämligen ”Hot Pink”, en skrikande rosa färg som blir synlig då man markerar t.ex. text på sidor. (HMTL5 Boilerplate 2013b) (MarakanaTechTV 2011)

2.3 Nedladdning av H5BP-paketet

För att starta ett H5BP-projekt går man in på hmtl5boilerplate.com och laddar ner ett paket. I skrivande stund är version 4.1.0. den nyaste versionen av H5BP och den publicerades den 21 januari 2013. På landningssidan kan man antingen välja att ladda ner den fulla versionen med allt H5BP-innehåll eller utgå från en anpassad version. De anpassade versionerna finns tillgängliga via Initializr, som är en mallgenerator. Hos Initializr kan man välja vilka delar av H5BP man vill eller inte vill ha med i paketet, beroende på hurudant projekt man har. (Initializr 2013) Källkoden för de separata filerna finns också direkt tillgänglig på github.com/h5bp/html5-boilerplate, om man endast önskar använda snuttar av Boilerplate.

2.3.1 Fulla versionen

Den fulla versionen innehåller ett fullständigt H5BP-paket och lämpar sig för alla tänkbara webbprojekt. Ladda ner zip-paketet från H5BP:s startsida och packa upp filerna. Strukturen på paketet ser ungefär ut som nedan i Figur 2. Innehållet behandlas noggrannare i kapitel 2.4.



Figur 2. Exempelstruktur på H5BP-paket (kopia från html5boilerplate.com)

2.3.2 Anpassade versioner

Via Initializr är det möjligt att generera anpassade versioner av H5BP. På denna sida kan man välja vilka delar man vill ha med i Boilerplate-paketet. Som utgångspunkt kan man välja den klassiska strukturen, en mall lämpad för en responsiv design eller en mall lämpad för webbprojekt i Bootstrap-ramverket utvecklat av Twitter. Efter att man valt utgångspaket, kan man ytterligare kryssa specifikt i de delar man behöver ha med. Man kan också välja att ladda ner en förenklad version av H5BP. Denna version innehåller inga kommentarförklaringar i koden. Detta utgångssätt är dock mera att rekommendera åt utvecklare som redan är bekanta med standardinnehållet i H5BP. (Initializr 2013)

2.4 Innehållet i H5BP

I följande kapitel kommer vi att dyka djupare in i H5BP:s struktur och innehåll samt se vad det fullständiga paketet har att erbjuda. Strukturen som behandlas relaterar till den presenterad i Figur 2.

2.4.1 index.html - märkspråksfilen

Detta är huvudfilen i mallen och webbsidans ”skelett”. Här byggs ramen för det kommande webbprojektet och härifrån laddar webbläsaren det material som ska visas utåt. Fastän mallen heter HTML5 Boilerplate är det inte nödvändigt att skriva HTML5-kod.

Dagens webbläsare förstår att rendera kod, så länge det finns en *doctype* i projektet. Index-filens första rad har den enkla HTML5 doctypen `<!DOCTYPE html>` som rekommenderas för standardrendering (w3.org 2013). Om man planerar att använda ett annat märkesspråk än HTML5 t.ex. XHTML kan det vara skäl att byta den.

Äldre webbläsare, speciellt äldre versioner av Microsoft Internet Explorer (IE), ställer ofta till med problem för webbutvecklare. Trots att de är föråldrade, är de tråkigt nog vanliga i användning och saknar stöd för den nya generationens webbt tekniker. H5BP har crossbrowser-stöd. Det betyder att koden som finns tillgänglig ska fungera i de flesta webbläsare. För webbläsare som saknar stöd för någon funktion presenteras villkorlig kod. Detta möjliggörs med villkorstagggar som riktar sig på föråldrade webbläsare.

```
1 <!DOCTYPE html>
2 <!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
3 <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
4 <!--[if IE 8]> <html class="no-js lt-ie9"> <![endif]-->
5 <!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
6 <head>
7 <meta charset="utf-8">
8 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
9 <title></title>
10 <meta name="description" content="">
11 <meta name="viewport" content="width=device-width">
12
13 <!-- Place favicon.ico and apple-touch-icon.png in the root directory -->
14
15 <link rel="stylesheet" href="css/normalize.css">
16 <link rel="stylesheet" href="css/main.css">
17 <script src="js/vendor/modernizr-2.6.2.min.js"></script>
18 </head>
```

Figur 3. Raderna 1-18 i *index.html* (skärmbild från Notepad++)

De första fyra raderna efter *doctype* (Figur 3) består av så kallade Internet Explorer villkorskommentarer som riktar specifika stilklasser åt elementet *html* för respektive versioner av IE. Orsaken till att dessa kommentarer finns så högt uppe och riktar sig på *html*-taggen, är för att de på det sättet undviker webbläsare att ladda fler stildokument än de verkligen kommer att använda. IE 6 kommer t.ex. att fastna för taggen `[if lt IE7]` och utsätts således att i stället använda klasserna *lt-ie9 lt-ie8 lt-ie7* medan IE 9 `[if gt IE8]` använder standardklasserna, som den klarar av (Irish 2008). Klassen *no-js* används

av Modernizr som är ett JavaScript-bibliotek och testar stödet för HTML5-teknologier. För varje funktion som stöds ändrar Modernizr klassen till *js* och webbläsaren renderar därefter. Modernizr presenteras mer omgående i kapitel 2.4.3.

Om vi ser på innehållet innanför *head*-taggen stöter vi först på ett par *meta*-taggar. Den första definierar teckenuppsättningen på vanligt HTML5-vis med *charset="utf-8"* (w3.org 2013b) (Mozilla Developer Network 2013). Den andra är däremot mera speciell och har som uppgift att tilldela Google Chrome Frame (GCF) åt äldre versioner av IE. GCF är en tilläggsdel för Internet Explorer och gör så att en föråldrad version av IE fungerar precis som en modern version av Chrome. *Meta*-taggen upptäcker i fall GCF finns installerad i IE och aktiveras därefter automatiskt. Alternativt kan man med hjälp av taggen vidarestyra användaren till att installera tilläggsdelen. Om serverkonfigurering används, rekommenderas det att man laddar denna meta-data ur serverkonfigureringsfilen (se kapitel 2.4.4). Dessa två *meta*-taggar är med flit placerade innan *title*-taggen för att undvika ett säkerhetsproblem. Efter *title*-taggen finns ytterligare två viktiga *meta*-taggar. Den ena är *description*, och en beskrivning av sidan som används av sökmotorer. Den andra är *viewport* som styr hur sidan laddas i mobila enheter, dvs. sidbredden optimeras till skärmstorleken. Efter dessa *meta*-taggar är det fritt fram att skriva egna *meta*-taggar. (Google Developers 2012) (HTML5 Boilerplate 2013b)

Nästa bit innanför *head*-taggen finns till för att de externa referenserna. Stilmallarna länkas på det moderna HTML5-sättet och som första har vi *normalize.css* som är en nollställningsfil (se kapitel 2.4.2.). Den andra är *main.css* som finns till för dina egna stilar och innehåller bl.a. hjälpklasser och Media Queries för responsiv design. Det rekommenderas att man använder enbart dessa stilmallar, eftersom fler referenser skulle betyda fler HTTP-förfrågningar och därmed långsammare sidladdning. Efter detta laddas Modernizr-biblioteket som är en JavaScript-fil (se kapitel 2.4.4).

```

19 <body>
20 <!--[if lt IE 7]>
21 <p class="chromeFrame">You are using an <strong>outdated</strong> browser. Please <a href="http://browsehappy.com/">upgrade your browser</a> or <a
href="http://www.google.com/chromeFrame/?redirect=true">activate Google Chrome Frame</a> to improve your experience.</p>
22 <![endif-->
23
24 <!-- Add your site or application content here -->
25 <p>Hello world! This is HTML5 Boilerplate.</p>
26
27 <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
28 <script>window.jQuery || document.write('<script src="js/vendor/jquery-1.10.1.min.js"></script>')</script>
29 <script src="js/plugins.js"></script>
30 <script src="js/main.js"></script>
31
32 <!-- Google Analytics: change UA-XXXXX-X to be your site's ID. -->
33 <script>
34     var _gaq=[['_setAccount','UA-XXXXX-X'],['_trackPageview']];
35     (function(d,t){var g=d.createElement(t),s=d.getElementsByTagName(t)[0];
36     g.src="//www.google-analytics.com/ga.js";
37     s.parentNode.insertBefore(g,s)}(document,'script'));
38 </script>
39 </body>
40 </html>

```

Figur 4. Raderna 19-40 i *index.html* (skärmbild från Notepad++).

Raderna 19-40 som omfattar resten av *index.html*-filen består av *body*-taggens innehåll och såklart *html*-sluttaggen. Innehållet i *body* börjar med en likadan IE-villkorskommentar som vi träffade på redan i sidhuvudet. Denna gång vill man rikta sig åt webbläsare äldre än IE 7 och visa en skriftlig uppmuntring om att deras användare borde ta och antingen uppdatera sina webbläsare eller installera Google Chrome Frame. Om man ämnar stöda IE 6 på sin sida, kan det vara skäl att radera denna rad. Utrymmet som följer är avsett för din egen kod.

På rad 27 laddas jQuery-biblioteket med en *script*-tag och som källkodsreferens har man valt den på Google CDN. URL-referensen är protokollfri utan `http://` eller `https://` för optimerat stöd. Google CDN är vald med omsorg, eftersom den är vanlig och kan därför redan finnas cachad i användarens webbläsare. Den är därför också prioriterad för att den troligtvis hämtas snabbare än en jQuery-version placerad på den egna servern. För offline-bruk och för ”fallback”-stöd har man dock även valt att länka till en lokalt tillgänglig version på följande rad, med hjälp av ett JavaScript-villkor. De nästa *script*-taggarna (rad 29-30) hämtar *plugin.js* som är menad för eventuella JavaScript-tilläggsdelar man kommer att använda och *main.js* som är menad för egna skript. (Marian 2012 s. 22 ff.)

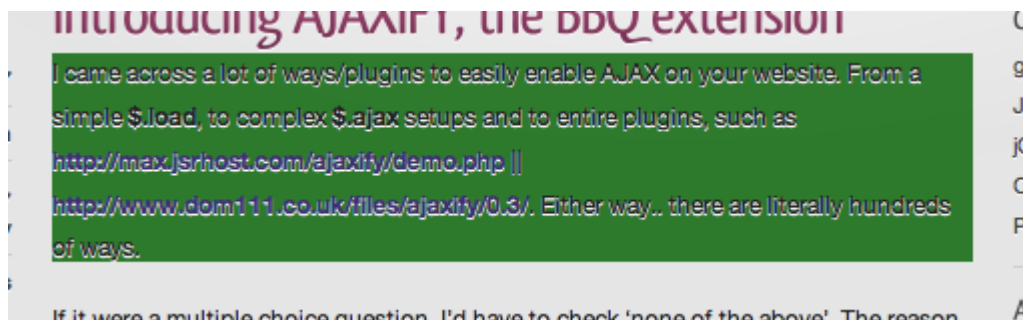
Längst ner på raderna 33-38 innan *body*- och *html*-slutttaggen finns en färdig Google Analytics-kodsnuttt placerad, färdig att tas i bruk. Det enda man som utvecklare behöver göra är att ersätta UA-XXXXX-X, med sin egen sidas unika identifikationskod. Man har valt att placera Google Analytics-kodsnutten längst ner på sidan så att metriken skickas först efter att sidan laddats helt och hållet. (Google Analytics 2013)

2.4.2 CSS – stilkonfigureringsfilerna

HTML5 Boilerplate använder sig av CSS-normalisering och andra användbara standardstilar. Paketet innehåller några vanliga hjälpklasser och utskriftstilar. I H5BP följer man en så kallad ”Mobile First” (mobil först) -utvecklingsfilosofi när det gäller design och bl.a. ett antal förjusterade Media Queries finns färdigt tillgängliga i CSS:en. Filerna som följer med paketet är *main.css* och *normalize.css*. (HTML5 Boilerplate 2013c)

Main.css innehåller överst en handfull H5BP-standarder som bygger på CSS-normalisering och finns till för att främja användarvänligheten. Till dessa hör:

- Typografiska inställningar för att förbättra text-läsbarheten. Exempelvis är normal textfärg förinställd till mörkgrå som är ögonvänligare än svart. Också länkar har setts över så att webbläsarnas olikheter gällande fokuseringsstilar är lika.
- Borttagning av oönskad textskugga vid val av text. Se exempel nedan.



Figur 5. Exempel på oönskad textskugga (hämtad från twitter.com/miketaylr/status/12228805301 13.5.2013)

- Som standard följer här även med ”Hot Pink”-markeringsfärgen, som blivit något av HTML5 Boilerplates varumärke.
- Finjusteringar för standardplacering av bilder, formulärfält och textareor. T.ex. lodrät placering av formulärfält i samband med label-taggar har finjusterats.

- Stil för en Google Chrome Frame frågeruta.

Efter dessa standarder finns det ett tomt utrymme för utvecklarens egen CSS-kod. För att undvika onödiga HTTP-förfrågningar och därmed nedsaktande av sidornas laddningshastighet, är det meningen att man skall klara sig med dessa två CSS-filer i sitt webbprojekt. Efter utrymmet för den egna CSS-koden följer några användbara hjälpklasser som styr hur element visas på sidan. Hjälpklasserna är nogt testade och fungerar genom alla webbläsare:

- **.ir**-klassen kan användas för text som vill ersättas med bild (image-replacement). Denna bilds referens bör då placeras in i själva stilkonfigureringen.
- **.hidden**-klassen kan användas för att gömma element totalt, även för skärmläsare om t.ex. innehållet senare ska fyllas med JavaScript. Skärmläsare är hjälpmedelprogramvara för bl.a. synskadade. (Wikipedia 2013)
- **.visuallyhidden**-klassen kan användas för att gömma text från webbläsare men hålla det synligt för skärmläsare. Denna klass är behändig då man vill förse skärmläsare med extrainformation.
- **.clearfix**-klassen kan användas på ett föräldraelement för att försäkra att dess barnelement alltid fysiskt befinner sig inom föräldraelementets ramar om de har ställts in att flyta (float), och förhindrar således oönskat överflöde (overflow). (Gallagher 2011)

För att underlätta responsiv webbdesign och för ”Mobile First” -tillvägagångssättet har man valt att placera några färdiga Media Queries i *main.css*. Trots dessa färdiga ramar, understryker H5BP-utvecklarna ändå att man bör modifiera ramarna enligt sina egna behov och resultat. Det finns så att säga inga perfekta klara lösningar.

Längst nere i *main.css* finns fördefinierade utskriftstilar. Utskriftstilarna aktiveras vid utskrift automatiskt då stilinnehållet innanför Media Query -taggen `@media print{...}` tas i bruk. Det är ofta vettigt att simplificera utseendet av webbsidan vid utskrift för att bevara sidans läsbarhet. I H5BP tar man bl.a. därför bort sidans eventuella bakgrundsfärg, rensar alla texteffekter och understryker länkar. Normaltvis hade man kanske län-

kat till en skild stilfil från sidans *head*-del, men pga. att webbläsare oberoende laddar stilfilen vare sig någon utskrift sker eller inte, har man i H5BP valt att lägga allt i samma CSS-fil för att undvika multipla HTTP-förfrågningar (Stoyan 2007).

Normalize.css är en modern HTML5-färdig CSS-nollställningsfil (Gallagher 2012). Med nollställning syftar man på utjämning av visuella renderingsolikheter som HTML-element av standard har i olika webbläsare, dvs. sådana element som man inte separat angett stilar för. Man vill på så sätt göra sidorna modernare och hållbarare för olikheter som finns i webbläsarna. Detta har också en stor betydelse för crossbrowser-stödet. Normalize.css är utvecklat av Nicolas Gallagher i samarbete med Jonathan Neal och är ett från H5BP skilt projekt och som man valt att inkludera pga. dess styrka och användbarhet. Styrkan kommer från att Normalize.css stöder en bred skara webbläsare, även mobila- och innehåller CSS som normaliserar HTML5-element, typografi, listor, inbäddat innehåll som media av olika slag, formulär och tabeller. Normalize.css är en liten CSS-fil men lika som Boilerplate-projektet har denna utvecklats till följd av hundratals timmar undersökning av webbläsares olikheter och brister men också styrkor. Man har valt att hålla *normalize.css* i H5BP som en separat fil för att den lätt skall kunna uppgraderas då en ny version dyker upp. För ideal prestanda skulle man givetvis i produktionsögonblicket kunna slå samman *normalize.css* med *main.css* och på så sätt endast referera till en stilkonfigureringsfil. Normalize.css mål är följande:

- Behålla användbara webbläsarstandarder, till skillnad från att radera dem
- Normalisera stilar för en bred skara HTML-element
- Rätta buggar och vanliga inkonsekvenser i webbläsare
- Förbättra användbarheten genom små förbättringar
- Förklara koden genom kommentarer och detaljerad dokumentation

2.4.3 JS - JavaScript- och jQuery-filer

H5BP-paketet kommer med en katalog *js* som innehåller två färdiga JavaScript-filer *main.js* och *plugins.js* samt underkatalogen *vendor*. *Main.js* är en fil som kan användas för utvecklarens egen JavaScript-kod. *Plugins.js* är avsedd för tredje part kod som man ämnar ansluta till sitt projekt, till exempel olika jQuery plugins. Hur som helst, om det

är frågan om en större, tyngre plugin med begränsat användningssyfte, kan det vara skäl att placera denna i en skild fil, och på så vis undvika onödig belastning av *plugin.js*. Båda filerna kan vara tillräckliga för små sidor, men såklart i utveckling av större sidor behöver man eventuellt överse behovet av fler JavaScript-filer. I början står båda filerna tomma med undantag från *plugin.js*. *Plugin.js* innehåller en kort funktion som undviker JavaScript-fel i IE 6 och IE 7 om man felsökt sina skript med konsolloggsfunktionen (*console.log*) men glömt kvar den. Med *log*-funktionen i *plugin.js* loggas uppgifterna endast i webbläsare som verkligen stöder konsolloggsfunktionen.

```
1 // Avoid `console` errors in browsers that lack a console.
2 (function() {
3     var method;
4     var noop = function () {};
5     var methods = [
6         'assert', 'clear', 'count', 'debug', 'dir', 'dirxml', 'error',
7         'exception', 'group', 'groupCollapsed', 'groupEnd', 'info', 'log',
8         'markTimeline', 'profile', 'profileEnd', 'table', 'time', 'timeEnd',
9         'timeStamp', 'trace', 'warn'
10    ];
11    var length = methods.length;
12    var console = (window.console = window.console || {});
13
14    while (length--) {
15        method = methods[length];
16
17        // Only stub undefined methods.
18        if (!console[method]) {
19            console[method] = noop;
20        }
21    }
22 }());
```

Figur 6. Funktion i *plugin.js* som undviker JavaScript-fel i webbläsare utan konsol (skärmbild från Notepad++).

Underkatalogen *vendor* finns till för tredje part bibliotekskod. På förhand finns här placerade en minimerad version av jQuery-biblioteket och Modernizr. Referensen till jQuery-biblioteket finns nere i *index.html* och det är frågan om en reservkopia som används om online-biblioteket av någon orsak inte laddas eller om man till exempel utvecklar sidan offline.

Modernizr är ett JavaScript-bibliotek som upptäcker HTML5- och CSS3-teknologier i slutanvändarens webbläsare. Med hjälp av Modernizr kan man enkelt skriva villkorlig JavaScript-kod för webbläsare, beroende på om de har eller saknar stöd för någon viss modern funktion. Modernizr körs snabbt vid sidladdning och skapar ett JavaScript-objekt med upptäckta resultat. Därefter adderas klasser enligt behov till html-elementet som utvecklaren styr genom att skriva sakenlig CSS. Till exempel om en webbläsare saknar stöd för runda kanter (*border-radius*), ger Modernizr elementet klassen *no-borderradius*, medan den för webbläsare med stöd ger elementet klassen *borderradius*. Modernizr kör ett dussin olika test och inkluderar vid behov *YepNope.js* som laddar endast de *.js*- och *.css*-resurser som slutanvändarens webbläsare behöver (YepNope 2013). Standardversionen av Modernizr som följer med H5BP kan om önskas också ersättas av en anpassad version, med endast de funktioner som utvecklaren anser sig behöva. (HTML5 Boilerplate 2013d)

2.4.4 .htaccess - Apache HTTP serverkonfigureringsfilen

Ursprungligen följer Apache-serverns konfigureringsfil *.htaccess* (hypertext access) med H5BP-paketet. Denna fil är packad med en mängd förhandskonfigureringar och inställningar för att öka prestandan och säkerheten på nätsidorna. För att utnyttja och förstå *.htaccess*-filens funktionalitet gäller det att bekanta sig med dokumentationen för den som finns tillgänglig både i H5BP-paketets medföljande *doc*-katalog och på H5BP:s hemsida. I dokumentationen finns instruktioner för förbättring av säkerhet, G-zip komprimering, cache-förföljning och hantering av URL-omdirigeringar som alla styrs från denna serverkonfigureringsfil. Pga. olikheter i underliggande utvecklarsystem och tidsbrist väljer jag ändå att inte presentera dessa konfigureringar här. För utvecklare som arbetar på andra serversystem än Apache finns det också tillhanda konfigurationsfiler och omgående dokumentation på H5BP:s hemsida. (HTML5 Boilerplate 2013e)

2.4.5 robots.txt – kontroll över webbrobotindexering

Webbrobotar är program som automatiskt genomkorsar webben och indexerar dess innehåll för t.ex. sökmotorer som Google. Med *robots.txt* kan man instruera dessa robotar och få dem att exkludera sidor från indexeringen, dvs. undvika att särskilda

sidor dyker upp i sökresultaten. Webbrobotar letar efter och läser automatiskt *robots.txt*-filen i nätsidans rotkatalog följer sedan ett så kallat exkluderingsprotokoll. Man styr robotarna med två enkla kommandon *User-agent* och *Disallow*. Om man t.ex. vill gömma filen *test.html* för alla sökmotorer kan man göra det med följande kommandon:

```
User-agent: *  
Disallow: /test.html
```

Figur 7. Exempelkodsnutt i *robots.txt* (skärmbild från Notepad++).

2.4.6 Övrigt

Förutom huvuddelarna presenterade tidigare följer det också med ett antal övriga behändiga filer i H5BP-paketet:

- En snyggare och anpassad *404.html* –felsida som kan modifieras enligt önskemål för att passa den egna stilen.
- *humans.txt* är en frivillig sida i rotkatalogen och något speciellt för H5BP. Sidan finns till för att presentera utvecklarna av sidan och kan vara intressant speciellt för andra utvecklare (humanstxt.org).
- I rotkatalogen finns placerat en färdig *favicon*, dvs. ikonerna som syns uppe vid adressfältet och på bokmärken i webbläsaren. Därtill finns det en bunt *apple-touch-icons* som används av Apple- men också Android-apparater då man sparar bokmärken på hemrutan. Dessa är givetvis möjliga att byta till egna ikoner.
- *crossdomain.xml* är en fil m.h.a. vilken man kan bevilja klientprogram såsom Adobe:s Flash Player tillstånd att hämta data tvärs över olika domain. (Adobe 2010)
- *readme.md* är en licensinformationsfil och innehåller även listinformation om de olika funktionerna i projektet och var man kan bekanta sig mera med dem.
- *gitattributes*- och *gitignore*-filer för Git-versionshantering finns inkluderade.

Också en tom katalog *img* finns placerad i rotkatalogen där det är meningen att placera alla bilder man har med i sitt projekt. I *doc*-katalogen, som nämnts tidigare finns all dokumentation för att förstå H5BP-filerna.

3 SKAPANDET AV EN BOILERPLATE-SIDA

3.1 Beskrivning av sidan

Jag har skapat en modellsida där jag utgått från HTML5 Boilerplate -mallen. Som mål hade jag att skapa en enkel responsiv bloggsida. Kravet för bloggsidan var att den skulle vara så mobilvänlig som möjligt, dvs. ideal också för visning på apparater med små skärmar. I utvecklandet utnyttjade jag några av H5BP:s funktioner och mot slutet testade jag sidans funktion. Följande kapitel är en beskrivning på hur jag gick till väga och vilka tekniker jag använde.

3.1.1 Responsiv webbdesign och Mobile First

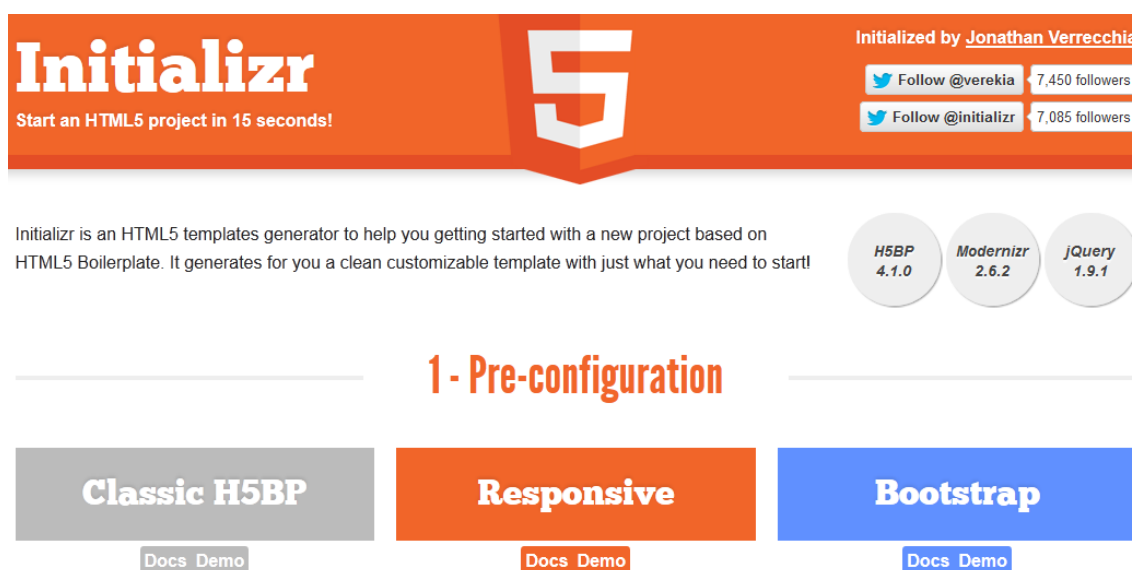
Responsiv webbdesign har utvecklats snabbt ända sedan Ethan Marcotte presenterade termen tre år sedan. Sedan dess har nya tekniker dykt upp, blivit bästa praxis och blivit del av vår ständigt föränderliga metodologi (Moore 2013). Grundpelarna i responsivt tänkande finns ändå kvar. En responsiv sida innebär en sida med en layout som anpassar sig till skärmen och apparaten den visas på. Exempelvis är texten alltid läsbar, bilderna i anpassad storlek och horisontellt rullande är inte nödvändigt. Detta möjliggörs genom CSS3:s Media Queries men flera faktorer bör tas i beaktande i designandet av ramen, bilderna och utseendet. Den viktigaste poängen är att i stället för absoluta pixlar använda relativa enheter som procenter och em. (Marcotte 2011)

Mobile First är en utvecklingsfilosofi och ett tillvägagångssätt där man designar först för mobilen. Mobile First innefattar också det som anses vara bästa praxis för mobil användarvänlighet. Luke Wroblewski har skrivit en hyllad strategisk guide om detta ämne (Wroblewski 2011). I guiden framkommer det bl.a. hurudant mobilanvändarens beteendemönster är, hur allt viktigt innehåll ska visas klart på sidan och hur stora länkar borde vara för att kunna väljas med ett finger. Mobile First och Responsiv webbdesign går mycket hand i hand och kompletterar varandra. Det är inte till exempel ovanligt att man bygger upp en responsiv design så att man först gör en design för mobil. På så sätt säk-

rar man att man får med allt det nödvändiga och att slutresultatet blir bra för mobil. I mitt modellprojekt har jag utnyttjat bägge ovannämnda tekniker.

3.2 Förberedelser och Initializr-botten

Boilerplate går att använda med vilket kodredigeringsprogram som helst. I detta modellprojekt använde jag Adobe Dreamweaver CS6. Som utgångspunkt till modellsidan använde jag den responsiva botten av Boilerplate som finns tillgänglig på Initializr.com (Figur 8). På Initializr kryssade jag i de delar av H5BP jag valde att ha med (se Figur 9). Eftersom det var frågan om en modellsida utan något slutanvändningssyfte, bestämde jag mig för att inte inkludera Google Analytics -kodsnutten eller Adobe Cross Domain -filen. .htaccess-filen tog jag med för att den har bra grundinställningar. Några ytterligare konfigurationer i serverkonfigureringsfilen valde jag att inte göra pga. tidsbrist och projektets storlek.



Figur 8. Steg 1, Vi väljer en responsiv botten på Initializr.com. (skärmbild)

2 - Fine tuning

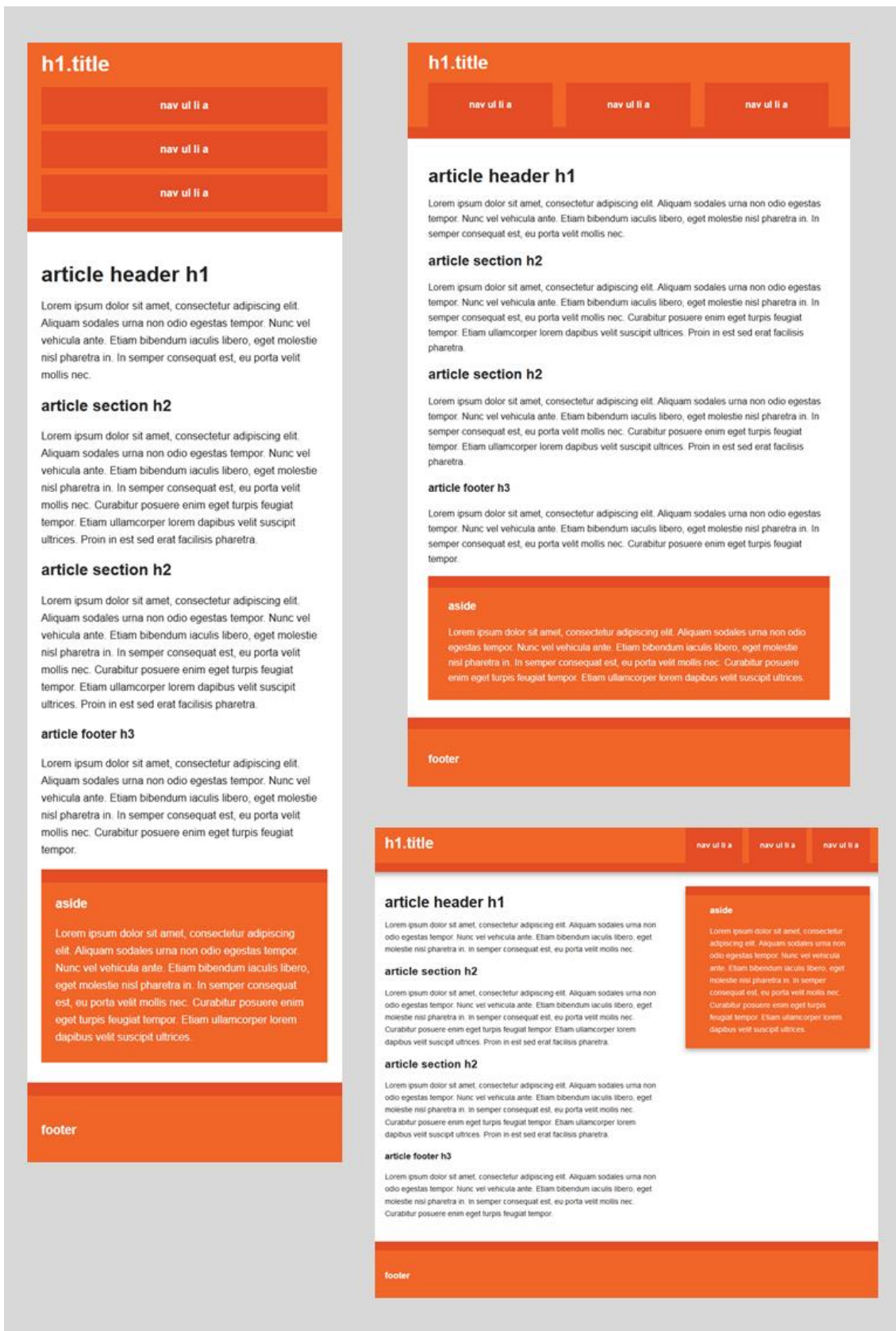
HTML/CSS Template	HTML5 Polyfills	jQuery
<input type="radio"/> No template	<input checked="" type="radio"/> Modernizr	<input checked="" type="checkbox"/> Minified
<input checked="" type="radio"/> Mobile-first Responsive	<input type="radio"/> Just HTML5shiv	<input type="checkbox"/> Development
<input type="radio"/> Responsive Bootstrap 2.3.0	<input checked="" type="checkbox"/> Respond - <i>Alternatives</i>	

H5BP Optional		
<input checked="" type="checkbox"/> IE Classes	<input checked="" type="checkbox"/> Favicon	<input checked="" type="checkbox"/> Humans.txt
<input checked="" type="checkbox"/> Chrome Frame	<input checked="" type="checkbox"/> Apple Touch Icons	<input checked="" type="checkbox"/> 404 Page
<input type="checkbox"/> Google Analytics	<input checked="" type="checkbox"/> plugins.js	<input type="checkbox"/> Adobe Cross Domain
<input checked="" type="checkbox"/> .htaccess	<input checked="" type="checkbox"/> Robots.txt	

Download it!	What's inside?
------------------------------	--------------------------------

Figur 9. Steg 2, där jag ytterligare väljer vilka delar jag vill ha med i botten. (skärmbild från Initializr.com)

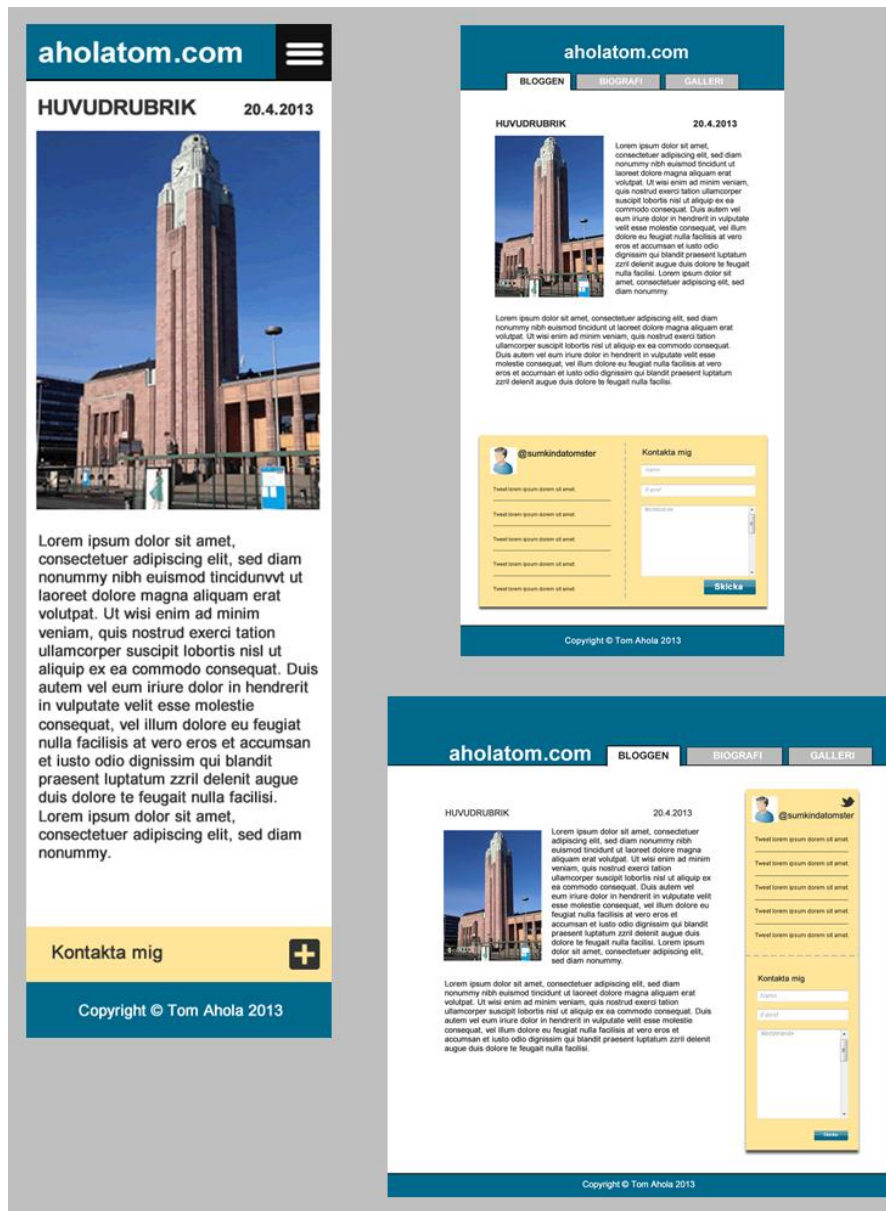
Efter att zip-paketet laddats ner, hämtar man ut filerna. Strukturen och innehåller i paketet känner vi till från kapitel 2 men till skillnad från en vanlig H5BP-mall, kommer denna med en färdig Mobile First- och responsiv-botten. Med andra ord finns det en del färdigt märkspråk i *index.html* och tillhörande CSS-kod i *main.css*. Det att botten är Mobile First, innebär att designen i förstahand är gjord med tanke på mobilenheter och små skärmstorlekar. Därefter har man stegvis designat responsivt mot större skärmstorlekar. En responsiv design använder en flexibel grid med procent- och em-enheter i stället för absoluta och oanpassbara pixlar. Denna flexibla grid anpassar sig i sin tur till skärmstorleken den visas på och styrs med hjälp av Media Queries i CSS-koden. Figur 10 demonstrerar hur *index.html* ser ut i en webbläsare utan några gjorda ändringar. När vi stegvis minskar och förstorar webbläsarfönstret, ser vi hur sidan anpassar sig och presenteras i olika layout. Till vänster har vi en tydlig och enkel mobilversion i smal vy, där en meny är placerad som knappar under varandra. Till höger har vi en litet större version som lämpar sig för en smartphone i landskapsläge eller en pekplatta i porträttläge. Märk hur menyn har flutit ut till höger. Sist längst ner har vi den fullbreda versionen som visas exempelvis på en bred datorskärm. Där har den orangea spalten flutit till höger om huvudinnehållet.



Figur 10. Responsiv design på botten från Initializr.com (skärmbilder från webbläsaren).

3.2.1 Egen design

Trots att botten fungerar bra till mitt bloggändamål som sådan, ville jag absolut göra sidan personligare. Som grund, innan jag börjar skriva kod, gjorde jag upp en egen design. Jag gjorde designen i Adobe Fireworks CS6 för alla de tre layouter vi såg i kapitel 3.2 (Figur 11).



Figur 11. Egen design på modellsidan gjord i Adobe Fireworks.

I min design har jag skisserat upp ett personligare utseende på bloggsidan. I mobilvyn finns till skillnad från den ursprungliga layouten, en utfällbar navigation dold bakom

ikonen bredvid sidrubriken. Den gula spalten har jag populerat med en Twitter-kontroll samt ett kontaktformulär i den mellanstora- och stora layouten. Navigationen tänkte jag skulle presenteras i form av flikar där den aktiva sidan är vitmarkerad och de inaktiva gråa. I själva blogginlägget har jag placerat en bild som ska, beroende på layout, flyta till en lämplig position och visas i en lämplig storlek.

3.3 Placering av innehåll och kodskrivande

Med en klar design kunde jag gå tillbaka till mallen och börja arbeta. Jag började med *index.html*-filen. I sidans *head*-del fyllde jag i titeln för sidan, likaså meta-beskrivningen (Figur 12). Meta-datan innan titel-taggen, som gällde Google Chrome Frame kunde jag plocka bort, eftersom den skulle laddas från *.htaccess*. Efter detta fyllde jag i övrig information på sidan som länktexter till navigationen, rubriker och footer-information.

```
<head>
  <meta charset="utf-8">
  <title>Modellsida - aholatom.com</title>
  <meta name="description" content="En modellsida skapad med HTML5 Boilerplate och Initializr.">
  <meta name="viewport" content="width=device-width">
```

Figur 12. Titel och meta-beskrivning ifylld (skärmbild från Dreamweaver).

Jag studerade den färdiga CSS-koden i *main.css* och gjorde därefter strukturella och färgmässiga ändringar enligt min design. Det var viktigt att inse att den normala CSS-koden var den avsedd för mobil, eftersom botten var Mobile First. Medan jag kodade var jag tvungen att ha ett webbläsarfönster öppet i ett smalt läge och sedan stegvis expandera fönstret, för att se hur ändringarna påverkade resultatet i de bredare vyerna. De bredare vyerna styrdes av Media Queries med *only screen*- och *min-width*-definitioner enligt Figur 13:

```
@media only screen and (min-width: 480px) { ... }
```

Figur 13. En Media Query som riktar sig på skärmar bredare än 480px.

I min design hade jag föreställt mig en annorlunda navigation för mobil än den presenterad i ursprungliga designen. Jag ville ha en knapp som dolde navigationen, något som man numera ofta ser hos responsiva sidor. För detta hittade jag en intressant lösning som heter Responsive Nav som jag placerad i *plugin.js*. Det är en lätt plugin som fungerar med endast JavaScript och har fallback också för webbläsare utan JavaScript. Fallback-funktionen fungerar då med hjälp av Modernizr som tillhör H5BP-mallen. (Responsive Nav 2013)

Till slut lade jag in en Twitter-kontroll och skrev HTML-koden för ett kontaktformulär i den frihängande sidospalten. Med Media Queries ställde jag in deras beteende i olika vyer enligt min design. Dessa tillsammans med bilden och innehållet följer samma responsiva princip, dvs. de visas stundvis som enskilda block och stundvis flytande bredvid varandra.

3.4 Övriga inställningar

I H5BP:s rotkatalog fanns färdigt alla ikoner jag behövde för sidan i rätt storlekar. För att skapa de enkla ikonerna använde jag Adobe Photoshop CS6. Jag ersatte *favicon.ico* och alla *apple-touch-icon-*.png*-filer med mina egna ikoner utan att ändra namnen på dem (Figur 14).



Figur 14. En enkel ikon för min modellsida.

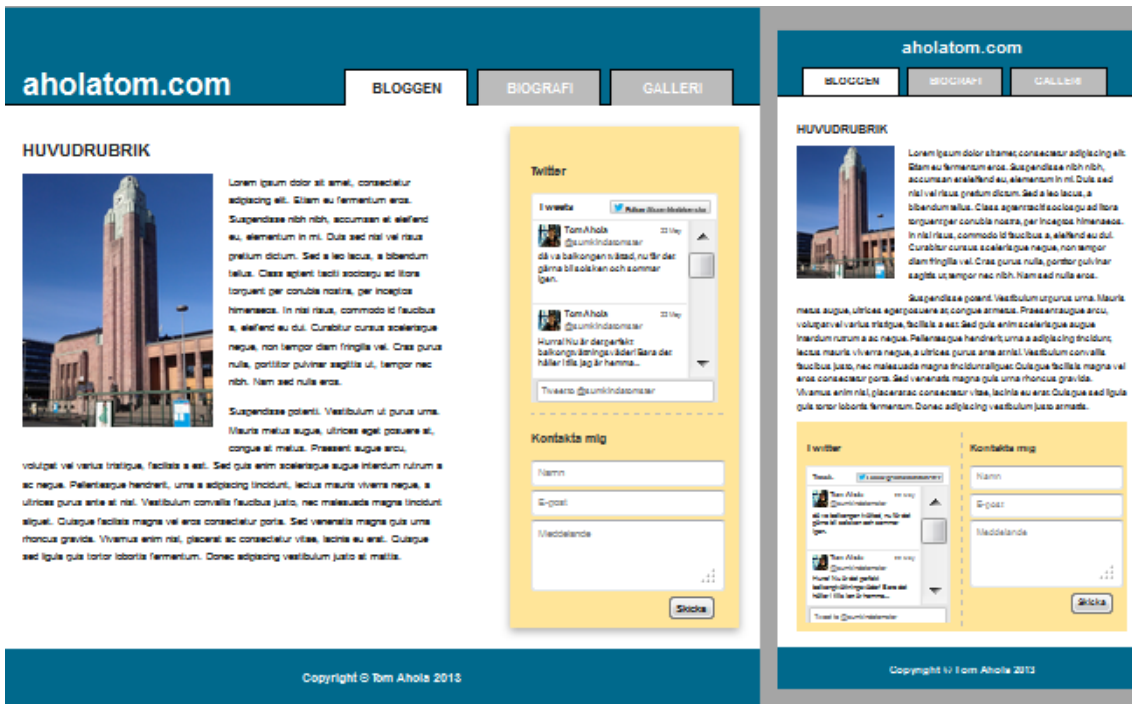
För detta projekt återstod ännu att fylla i *humans.txt*, d.v.s. berätta vem som gjort sidan och vilka tekniker som använts (Figur 15).

```
# humanstxt.org/  
# The humans responsible & technology colophon  
  
# TEAM  
  
  Web developer: Tom Ahola  
  Site: aholatom.com/modell  
  Twitter: @sumkindatomster  
  Location: Helsinki, Finland  
  
# TECHNOLOGY COLOPHON  
  
  HTML5, CSS3  
  Normalize.css, Modernizr
```

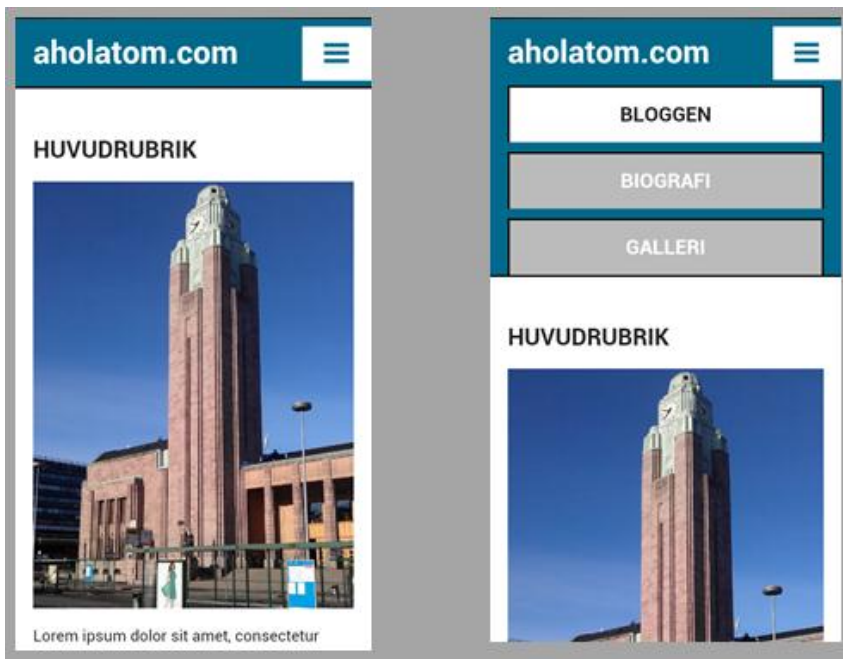
Figur 15. *humans.txt* ifylld med information om vem som skapat sidan och vilka tekniker som använts.

3.5 Resultat och tester

Den slutgiltiga modellsidan jag producerade finns att beskåda på aholatom.com/modell eller i Figur 16. Byggandet av sidan tillsammans med designarbetet tog ca tre arbetsdagar och en dag för testande och buggrättning. Jag blev tvungen att göra en del kompromisser och ändringar i designen efter att jag satte igång med kodarbetet. Till exempel hade jag ursprungligen planerat en CSS3–gradient som bakgrund, men på grund av webbläsarnas ojämna och otillräckliga stöd för funktionen, bestämde jag mig för att slopa idén. Sidan har testats i olika webbläsare och resultatet är åtminstone för skribenten, målsättningen och till kraven tillfredsställande. I sidan har bl.a. H5BP-hjälpklassen *clearfix* använts som till synes fungerade bra. Dock när JavaScript-kopplades bort i testerna märkte man att layouten inte höll. Det är alltså en fråga som webbutvecklaren bör ta ställning till, om man kan förutsätta JavaScript-stöd av slutanvändaren eller ej? En oroväckande detalj förblev, att sidan fungerade dåligt IE-webbläsare äldre än version 9. Skribenten har inte lokaliserat problemet i skrivande stund, men anser att det bör röra sig om någon liten detalj, eftersom både Initializr-botten och JavaScript-navigationen i enskilda tester fungerade bra. Modellsidans funktion i moderna webbläsare och mobila apparater kan ändå anses positiv, vilket var kravet. Som testapparatur har också en Android-baserad HTC Sensation smartphone och en Iphone 4 använts (Figur 17, Figur 18). Också utskriftsvyn har testats med god framgång.



Figur 16. Slutgiltiga modellsidan i webbläsare (skärmbild).



Figur 17. Slutgiltiga modellsidan testad med HTC Sensation -mobil (skärmbild).



Figur 18. Slutgiltiga modellsidan testad med Iphone 4 (skärmbild).

4 DISKUSSION

HTML5 Boilerplate är ett kraftfullt och bra verktyg för webbutvecklare. Jag märkte ändå att även om mallen är lätt att ta i bruk, så är dess omfattning och innehåll absolut ingen självklarhet. Dokumentationerna bör studeras noga för att man ska förstå Boilerplate och vad den gör. Endast på det sättet får man största möjliga nytta av mallen. Sedan hur djupt man väljer att studera, hur man kommit till de olika lösningarna och varför de representerar bästa praxis, är upp till en själv. HTML5 Boilerplate är ett startpaket för att skapa snabba och robusta sidor. HTML5 Boilerplate är god webbutvecklingspraxis, men det är ingen genväg. Erfarenheten jag tar med mig i och med detta arbete är enormt värdefull. Jag har inte bara lärt mig nya saker om webbutveckling jag har också lärt mig hurudan skribent jag är, och hur jag bäst arbetar.

Mitt arbete tar inte slut här. Jag har fått nya nycklar till nya dörrar och jag visste inte ens hur stor konstruktion jag gav mig in på. Det finns mycket att utforska ännu i HTML5 Boilerplate och det finns mycket att utforska ännu när det gäller god webbutvecklingspraxis. Till exempel skulle jag gärna vilja se hur Boilerplate fungerar på en större, even-

tuellt kommersiell sida. Jag skulle vilja studera och förstå vikten av bra serverkonfiguration. Det i sig självt, skulle vara ett eget examensarbete. Tyvärr nästan, kan man säga att det finns oändligt med bra tekniker, och man lär sig bara det man verkligen stöter på eller tar sig tiden att lära. I och med mobila apparaternas frammarsch har man märkt att man behöver se på webben ur alldeles nya perspektiv. Det är det som håller vår bransch intressant, den ständiga utvecklingen.

KÄLLOR

- Adobe. 2010, Cross-domain policy file specification, *Adobe Developer Connection* [www] (Uppdaterad: 22.1.2010)
Finns på: http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html
[Hämtad 16.5.2013]
- Gallagher, Nicholas. 2011, *A new micro clearfix hack* [www] (Uppdaterad: 21.4.2011)
Finns på: <http://nicolasgallagher.com/micro-clearfix-hack/>
[Hämtad 17.3.2013]
- Gallagher, Nicholas. 2012, *About normalize.css* [www] (Uppdaterad: 17.6.2007)
Finns på: <http://nicolasgallagher.com/about-normalize-css/>
[Hämtad 17.3.2013]
- Google Analytics. 2012, *Tracking Site Activity* [www] (Uppdaterad: 29.10.2012)
Finns på: <https://developers.google.com/chrome/chrome-frame/>
[Hämtad 12.3.2013]
- Google Developers. 2012, *Enable open web technologies in Internet Explorer* [www] (Uppdaterad: 18.9.2012)
Finns på: <https://developers.google.com/chrome/chrome-frame/>
[Hämtad 12.3.2013]
- H5BP.net. 2013, *An online showcase* [www] (Uppdaterad: 17.5.2013)
Finns på: <http://h5bp.net/>
[Hämtad 19.5.2013]
- H5BP forks. 2013 [www] (Uppdaterad: 12.4.2013)
Finns på: <https://github.com/h5bp/html5-boilerplate/wiki/forks>
[Hämtad 19.5.2013]
- H5BP sites. 2013 [www] (Uppdaterad: 16.5.2013)
Finns på: <https://github.com/h5bp/html5-boilerplate/wiki/sites>
[Hämtad 19.5.2013]
- HTML5 Boilerplate. 2012, *history* [www] (Uppdaterad: 30.8.2012)
Finns på: <https://github.com/h5bp/html5-boilerplate/wiki/history>
[Hämtad 20.4.2013]
- HTML5 Boilerplate. 2013a, *README.md* [www] (Uppdaterad: 30.1.2013)
Finns på: <https://github.com/h5bp/html5-boilerplate/tree/v4.2.0>
[Hämtad 23.4.2013]

- HTML5 Boilerplate. 2013b, *The HTML* [www]
Finns på: <https://github.com/h5bp/html5-boilerplate/blob/master/doc/html.md#the-order-of-meta-tags-and-title>
[Hämtad 13.5.2013]
- HTML5 Boilerplate. 2013c, *The CSS* [www]
Finns på: http://html5boilerplate.com/html5boilerplate-site/built/en_US/docs/css/
[Hämtad 15.5.2013]
- HTML5 Boilerplate. 2013d, *The JavaScript* [www]
Finns på: http://html5boilerplate.com/html5boilerplate-site/built/en_US/docs/js/
[Hämtad 17.5.2013]
- HTML5 Boilerplate. 2013e, *.htaccess*. [www]
Finns på: <https://github.com/h5bp/server-configs/tree/master/apache>
[Hämtad 11.5.2013]
- Initializr. 2013 [www]
Finns på: <http://www.initializr.com/>
[Hämtad 12.3.2013]
- Irish, Paul. 2008, *Conditional Stylesheets vs CSS Hacks? Answer: Neither!* [www]
(Uppdaterad: 17.1.2012)
Finns på: <http://paulirish.com/2008/conditional-stylesheets-vs-css-hacks-answer-neither/>
[Hämtad 10.3.2013]
- Manian, Divya. 2012, *HTML5 Boilerplate Web Development*, (s. 22 ff.),
Packt Publishing [e-bok]
ISBN 978-1-84951-850-5
- MarakanaTechTV. 2011, *Paul Irish on HTML5 Boilerplate* [www/video] (Uppladdad: 4.2.2011)
Finns på: <http://www.youtube.com/watch?v=qyM37XKkmKQ>
[Hämtad 12.3.2013]
- Marcotte, Ethan. 2011, *Responsive Web Design, A Book Apart*
ISBN 978-0-9844425-7-7
- Moore, Jordan. 2013, *Responsivle Considerations For Responsive Web Design*, Smashing Magazine [www] (Uppdaterad: 11.3.2013)
Finns på: <http://www.phpied.com/delay-loading-your-print-css/>
[Hämtad 17.3.2013]
- Mozilla Developer Network. 2013, *<meta>* [www] (Uppdaterad: 9.5.2013)
Finns på: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta>
[Hämtad 27.5.2013]

Responsive Nav. 2013 [www]

Finns på: <http://responsive-nav.com/>

[Hämtad 22.5.2013]

StatCounter. 2013, Mobile vs. Desktop in Finland, *StatCounter Global Stats* [www]

Finns på: http://gs.statcounter.com/#mobile_vs_desktop-FI-monthly-201204-201304

[Hämtad 22.5.2013]

Stoyan, Stefanov. 2007, *Delay loading your print CSS* [www] (Uppdaterad: 28.2.2012)

Finns på: <http://www.phpied.com/delay-loading-your-print-css/>

[Hämtad 17.3.2013]

w3.org. 2013a, HTML 5.1, kap. 8.1.1 *The Doctype* [www]

Finns på: <http://www.w3.org/html/wg/drafts/html/master/syntax.html#the-doctype>

[Hämtad 14.3.2013]

w3.org. 2013b, *Declaring character encodings in HTML* [www]

Finns på: <http://www.w3.org/International/questions/qa-html-encoding-declarations#html5charset>

[Hämtad 14.3.2013]

Wikipedia. 2013, *Screen reader* [www] (Uppdaterad: 10.4.2013)

Finns på: http://en.wikipedia.org/wiki/Screen_reader

[Hämtad 11.5.2013]

Wroblewski, Luke. 2011, *Mobile First, A Book Apart*

ISBN 978-1-937557-02-7

YepNope. 2013 [www]

Finns på: <http://yepnopejs.com/>

[Hämtad 17.3.2013]