

Web design haltuun

Manuaali www-sivujen toteuttamiseen

Jussi-Pekka Karisaari

Opinnäytetyö
Marraskuu 2014

Tietojenkäsittelyn koulutusohjelma
Luonnontieteiden ala





Tekijä(t) Karisaari, Jussi-Pekka	Julkaisun laji Opinnäytetyö	Päivämäärä 17.11.2014
	Sivumäärä 29 + 15	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: X
Työn nimi Web design haltuun Manuaali www-sivujen toteuttamiseen		
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma		
Työn ohjaaja(t) Immonen, Jarkko		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu		
Tiivistelmä <p>Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma tarvitsee uudenlaista opetusmateriaalia Web Page Development -opintojakson toteuttamiseen. Kyseisellä opintojaksolla opiskelijalle opetetaan www-sivujen kehityksen perusasiat käytetyistä työkaluista ja tekniikoista kokonaisten sivujen luomiseen.</p> <p>Tähän asti tietojenkäsittelyn koulutusohjelman opintojaksoilla on painotettu lähinnä kontaktiopetusta ja harjoitustehtäviin nojautuvaa opiskelua. Näin myös Web Page Development -opintojaksolla. Tulevaisuudessa opetuksen pääpainopistettä halutaan siirtää opiskelijoiden itsenäiseen työskentelyyn: heille pyritään antamaan enemmän vastuuta omasta opiskelustaan, ja näin ollen selkeän ja tehokkaan opintomateriaalin saatavuus tulee taata. Opetusmateriaaleja, jotka tukevat itsenäisen tiedonhauun ja tehtävienratkaisukykyjen kehitystä, täytyy tutkia ja kehittää.</p> <p>Tutkimuksen tehtävänä oli kehittää näiden tarpeiden pohjalta opiskelijoille osoitettu verkkomanuaali, jossa käydään vaiheittain läpi kokonaisen www-sivun toteuttaminen. Tehtävänannon ja opintojakson sisällön mukaisesti manuaalin materiaaleineen kuului olla sisällöltään selkeä ja perusasiat käsittävä. Siksi manuaalia varten luotu mallisivusto, jonka kehittämistä harjoitellaan, on yksinkertainen mutta käytännöllinen. Opiskelijan odotetaan pystyvän hahmottamaan www-sivun luomisen kokonaisprosessi annettuja vinkkejä ja ohjeita mukaillen. Tämän kaiken tarkoituksena on tukea WPD-opintojakson muuta opetusta.</p> <p>Tutkimuksen tuloksina saatiin paitsi uutta opintomateriaalia verkkomanuaalin muodossa, myös selkeä läpikatsaus opiskelijoille suunnatun aineiston tuottamisen prosessiin. Vastaavanlaista opintomateriaalin tuottamista kehittämistutkimuksen puitteissa voidaan hyvin soveltaa jatkossakin Jyväskylän ammattikorkeakoulun opetustyön kehittämiseen. Verkkomanuaalia voidaan myös laajentaa käsittelemään syventäviä www-suunnittelun teemoja.</p>		
Avainsanat (asiasanat) Www-kehitys, HTML, CSS, www-sivu, manuaali		
Muut tiedot		



Author(s) Karisaari, Jussi-Pekka	Type of publication Bachelor's thesis	Date 17.11.2014
		Language of publication: Finnish
	Number of pages 29 + 15	Permission for web publication: X
Title of publication Figuring out Web Design Tutorial for web page development		
Degree programme Business Information Systems		
Tutor(s) Immonen, Jarkko		
Assigned by JAMK University of Applied Sciences, Business Information Systems		
Abstract <p>JAMK University of Applied Sciences' degree programme for Business Information Systems needs new education material for Web Page Development course. In that particular course students are taught the basics of web page development, ranging from tools of the trade and appropriate techniques to creating a full web site.</p> <p>So far the education in Business Information Systems' courses has mostly based on contact lessons and regular exercises. This is also true for the Web Page Development course. In future the emphasis of teaching is aimed towards independent learning: students are to be given more responsibility over their own studying process, and therefore the availability of coherent and efficient education material must be guaranteed. These kind of education materials to support the development of independent information retrieval and troubleshooting skills must be further researched and developed.</p> <p>Based on these requirements, the mission of this research was to develop a web tutorial aimed for students of JAMK with step-by-step instructions on creating a web page. In accordance with the given assignment and the aim of WPD course the manual was to be coherent in content and comprising mainly the basics of web development. Thus the web page that is used as an example in the manual is also simple but practical. Students are expected to comprehend the full process of development of a web page based on any given tips and instructions. All of this will support the other teaching methods of WPD course.</p> <p>The results of the research included not only the new education material in a form of a web tutorial but also a comprehensive view on the process of production of the particular kind of material. This method of producing education material in the form of development project can further be applied to improving the teaching methods in JAMK University of Applied Sciences. The web tutorial can also be expanded to cover more advanced themes of web development.</p>		
Keywords/tags (subjects) web development, HTML, CSS, web page, tutorial		
Miscellaneous		

SISÄLTÖ

1	JOHDANTO	3
2	TUTKIMUSASETELMA	4
2.1	Toimeksiantaja	4
2.1.1	Web Page Development.....	4
2.1.2	Pedagogiset lähtökohdat	5
2.2	Tutkimusmenetelmä	6
2.2.1	Kehittämistutkimus	6
2.2.2	Tutkimuskysymykset	7
3	WEB-KEHITYS JA SEN TEKNIIKAT	7
3.1	Web-kehitys yleisesti.....	7
3.2	WWW-sivujen toiminta ja arkkitehtuuri	8
3.3	HTML	9
3.3.1	Tagit.....	11
3.3.2	HTML5	12
3.4	CSS.....	13
3.5	Työvälineet	16
3.5.1	Tekstieditori.....	16
3.5.2	Www-selain	16
3.5.3	FTP-palvelinohjelmat.....	17

3.5.4	Kuvankäsittelyohjelmat.....	17
4	TUTKIMUKSEN TOTEUTTAMINEN.....	18
4.1	Www-sivun layout.....	18
4.2	Siirtyminen koodiin	21
4.3	Manuaalin kirjoittaminen	23
5	POHDINTA.....	25
	LÄHTEET	27
	LITTEET	29

KUVIOT

	Kuvio 1. Internetin toimintaperiaate	9
	Kuvio 2. HTML:n pohjarakenne.....	10
	Kuvio 3. Sama HTML-dokumentti laajennettuna	11
	Kuvio 4. HTML-dokumentti. Huomaa h1- ja p-tagit	14
	Kuvio 5. CSS-dokumentti, joka antaa tyyliarvoja em. HTML-tiedostolle.....	15
	Kuvio 6. Sivun layout.....	20
	Kuvio 7. HTML-dokumentin valmiita elementtejä	22
	Kuvio 8. Em. HTML-dokumenttia täydennetään CSS:n avulla.	22
	Kuvio 9. Ote manuaalista	24

1 JOHDANTO

2010-luvun puolivälin lähestyessä World Wide Webin eli Internetin merkitys ihmisen elämässä on suurempi kuin koskaan. Lähestulkoon kaikki maailmalla liikkuva informaatio on kytköksissä Internetin valtavirtaan, ja tämä käsittää paitsi perinteiset kryptatut tietokoneiden välillä lähetettävät datapaketit, myös erilaiset median muodot, kuten verkossa jaettavat ja näytettävät kuvat ja videot. Kaukana ovat siis ne ajat, jolloin Internet oli vain rajatun kohderyhmän käyttämä sisäinen tiedonsiirtoväylä, kenen tahansa internet-yhteyden omaavan kyetessä nykyään selaamaan verkkoa täysin vapaasti.

Mitä enemmän informaatiota siirretään Internetiin, sitä suuremmassa merkityksessä tulee olemaan niiden ammattilaisten osuus, jotka hallitsevat www-arkkitehtuurin ja suunnittelun käytänteet. Keskivertoiselle Internetin käyttäjälle verkkotoiminnan tekniset asiat eivät merkitse juuri mitään, sillä vaikka käyttäjältä edellytetään entistä enemmän verkkolukutaitoa, niin valtaosan kosketus tähän tiedon ja datan valtaväylään on yhä selaimella toteutettavaa pintapuolista "surffailua". WWW-sivut toimivat rajapintana mutkikkaan koodikielen ja Internetin käyttäjän välillä verkkosuunnittelussa käytettävien koodikielten kääntyessä luettavaksi tekstiksi ja katseltaviksi kuviksi.

On siis ensiarvoisen tärkeää, että uusia www-suunnittelun osaajia löytyy tulevaisuudessaakin runsaasti. Parastahan asiassa on, että kuka tahansa voi aloittaa verkkosivujen tekemisen. Tätä tarkoitusta varten on jopa luotu monia blogien ja nettisivujen julkaisualustoja, jotka tarjoavat valmiiksi räätälöityjä työkaluja ja muotteja yksinkertaisen, henkilökohtaisen sivuston toteuttamiseen.

Vaativamman, persoonallisen sivuston suunnittelu vaatii kuitenkin kattavampaa käsitystä www:stä. Kaikki lähtee perusasioista, kaikkein alkeellisimmista verkkosivuista ja käytettävistä tekniikoista. Näitä lähtökohtia voidaan opetella ja soveltaa vaikkapa yksinkertaisen ja helppolukuisen manuaalin avulla.

Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma ohjaa opiskelijoita www-kehityksen saloihin Web Page Development -opintojakson puitteissa. Pedagogisesti kyseisen opintojakson opetus toteutetaan opiskelijan itsenäisen työskentelyn periaatteiden mukaisesti. Tämän opinnäytetyön tarkoitus on tuottaa opintojakson käyttöön manuaalityyppinen tuotos, joka tukee tätä itsenäisen työskentelyn lähtökohtaa.

2 TUTKIMUSASETELMA

2.1 Toimeksiantaja

Jyväskylän ammattikorkeakoulu (JAMK University of Applied Sciences) on vuodesta 1994 lähtien toiminut korkeakoulu, joka on lajissaan yksi Suomen suurimmista. JAMK tarjoaa vaihtelevasti erilaisia koulutusaloja sekä nuorten että aikuisten korkeakoulututkintojen tarjonnasta.

Tämän opinnäytetyön toimeksiantajana toimii ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma (TIKO). Se kouluttaa opiskelijoistaan ICT-alan asiantuntijoita, joiden osaamisalaksi lukeutuu muun muassa tietojärjestelmien kehitys ja sovellus, digitaalinen media sekä liiketoiminta. Vuosittain tietojenkäsittelyn koulutusohjelman opiskelun aloittaa n. 40 uutta opiskelijaa (Bisneksen ja tietotekniikan osaaja. N.d.).

2.1.1 Web Page Development

TIKOn opintojen ensimmäisenä vuonna painotetaan Web- ja sovelluskehityksen perusteita, joihin myös Web Page Development (WPD) -opintojakson sisältö lukeutuu. Kyseisellä opintojaksolla opiskelijoille esitellään www-sivun kehitykseen tarvittavia alkeistietoja, pääpainon ollessa HTML- ja CSS-koodikielten opiskelussa ja käytännön soveltamisessa perustyökaluja käyttäen. Muita käsiteltäviä teemoja ovat JavaScript-

koodikielen perusteet, ja myös yleisiä www-sivun suunnittelun niksejä, kuten käytettävyyttä ja visuaalisuutta, käydään aktiivisesti läpi. Opintojakso tulee toimimaan lähtöpisteinä laajemmalle web-kehitykselle. Opetuskielenä toimii englanti (Web Page Development -opintojakson informaationsivu 2014).

Opintojakso suoritetaan sekä luentojen tukemana lähiopetuksena että itsenäisenä etätyöskentelynä harjoitustehtävien avulla. JAMKilla ja sitä myöten tietojenkäsittelyn koulutusohjelmalla on käytössään verkkotyöympäristö Optima, johon kirjautumalla opiskelija pääsee opiskeluun tarvittaviin materiaaleihin ja harjoitustehtäviin käsiksi. Opintojakson päätyttyä opiskelijan odotetaan osaavan suunnitella ja luoda omin päin yksinkertainen staattinen www-sivu tehtävänantoa mukaillen opinto- ja/tai saatavilla olevaa verkkomateriaalia hyödyntäen (Web Page Development -opintojakson informaationsivu 2014).

2.1.2 Pedagogiset lähtökohdat

TIKOn opintojaksoilla ollaan pedagogisesti suuntautumassa yhä enenevässä määrin verkko-opiskeluun. Tämä tarkoittaa opiskelijan kannalta aktiivisempaa itseopiskelua kotona tai kouluympäristössä, samalla kun kontaktiopetusta karsitaan. Tällä pyritään edesauttamaan opiskelijan itsenäistä tiedonhakua ja tehtävienratkaisukykyä sen sijaan, että opintojakson opettaja opastaisi kädestä pitäen jokaista henkilöä kaikissa työn vaiheissa.

Edellä mainittu ei tarkoita sitä, että opiskelija jätettäisiin yksin ratkomaan opintojakson suoritteet ilman minkäänlaista apua. Päinvastoin, verkko-opiskelussa Internetistä löytyvän, ennen kaikkea laadukkaan opiskeluaineiston merkitys korostuu. Materiaalin täytyy olla räätälöity riittävän selkeäksi ja helposti lähestyttäväksi, mutta toisaalta sen kuuluu olla myös haastavaa.

Tästä kumpuaa tämän tutkimuksen tarve. Web Page Development -opintojakson opiskelijoille pyritään tarjoamaan parempaa opintomateriaalia, joka toivottavasti edesauttaa tehokasta itsenäistä opiskelua. Täten opinnäytetyön ohessa syntyvä ma-

nuuali tulee toimimaan paitsi käyttökelpoisena materiaalina, myös malliesimerkkinä tulevaisuudessa kenties käyttöön otettavasta koulutusmetodista.

2.2 Tutkimusmenetelmä

2.2.1 Kehittämistutkimus

Opinnäytetyö on tyypiltään kehittämistutkimus. Sen pääpainona on kehittää jotain valmista ideaa/menetelmää paremmaksi esitettyjen tarpeiden ja tunnistettujen haasteiden pohjalta (Kehittämistutkimuksesta saadaan käytännöllistä tietoa opetuksen kehittämiseen. 2013). Aiempaa ilmiötä ja siitä saatuja kokemuksia sekä siihen liittyvää teoriaa käydään läpi, ja tätä kaikkea käytetään hyväksi kehitystyön suunnitteluvaiheessa, jota seuraavat varsinaisen konkreettisen tutkimuksen toteuttaminen ja viimeisenä raportointi ja tutkimuksen yhteenveto (Kananen 2012, 47).

Tutkimuksen tuloksena saadaan yleensä jonkinlainen konkreettinen tuotos, vähintään kirjallinen esittely/raportti uudesta kehitetystä ideasta. Oppimisen kannalta tärkeäksi muodostuu myös itse tutkimuksesta kirjoitettu selonteko, joka voi toimia opettavana kokemuksena paitsi itse tutkijalle, myös muille aiheita tulevaisuudessa käsitteleville.

Toimeksiantajalta tulleen tiedonannon mukaan opiskelijan itsenäistä työskentelyä tukevaa materiaalia tarvitaan enemmän. Tämän nimenomaisen työn tarkoituksena on ensisijaisesti tuottaa uutta, hyödyllistä opetusmateriaalia WPD-opintojakson opiskelijoille ja tutkia, miten tulevaisuudessa voitaisiin taata vastaavan tehokkaan opetusmateriaalin saatavuus. Tutkimuksen tuloksena syntyy yksinkertaisen www-sivun luomiseen perehdyttävä opas, joka tulee toimimaan opetusmateriaalina WPD-opintojakson opiskelijoille.

2.2.2 Tutkimuskysymykset

Tutkimuksen toteuttamiseen johtavat kysymykset voidaan tiivistää, johon vastaamalla saadaan samalla vastaukset yleiskysymyksiin mitä tehdään, miksi tehdään ja miten tehdään:

- Millaisilla työvälineillä Web Page Development -opintojakson opiskelijoiden kokonaisnäkemysten kehittymistä voitaisiin tukea?

3 WEB-KEHITYS JA SEN TEKNIIKAT

3.1 Web-kehitys yleisesti

Web-suunnittelun perimmäisenä tarkoituksena on luoda käyttäjälle helposti lähestyttävä ja -käytettävä www-sivusto, joka on räätälöity käyttötarkoituksensa mukaisesti. Www-sivujen visuaalisuudella on sivuston toiminnan kannalta merkittävä rooli: paitsi että sivujen täytyy selkeiden ja tarkoituksenmukaisten kuvien ym. grafiikan avulla kuvastaa edustamansa toimijan imagoa, on tärkeää korostaa eri elementtien järkevää sijoittelua. Sivuston eri toiminnot, linkit ja informaatio täytyy kyetä löytämään vaivattomasti ja loogisesti, tietynlaisia prioriteetteja noudattaen. Esim. www-sivun yläosa pyhitetään lähes poikkeuksetta sivuston omistajan logolle ja sivustolla selaamista helpottavalle navigointipalkille, koska näillä halutaan herättää selaajan huomio ja kertoa tälle lyhykäisyydessään, mitä sivustolla on tälle tarjota. Tässä kaikessa on kyse *käytettävyydestä*, jota korostetaan Web-kehityksen alkeita opettavilla opintojaksoilla (Nielsen, J. 2012).

Web-kehitykseen kuuluu perinteisten verkkosivujen suunnittelun ohella käyttäjälle näkymättömien elementtien, esim. tietokantojen ja sovellusten, toteutus ja kontrollointi. Varsinaisen näkyvän www-sivun taustalla on paljon ominaisuuksia, jotka vaikuttavat olennaisesti sivuston toimintaan ja sille syötetyn datan tallentamiseen ja käsittelyyn. Monet näistä toiminnoista vaativat erillistä ohjelmoinnin osaamista, joka

poikkeaa monilta osin graafisesta verkkosivujen kehityksestä. Useimmiten www-sivujen kehitystä opiskelevat oppivat vähintään ohjelmoinnin alkeistaitoja.

3.2 WWW-sivujen toiminta ja arkkitehtuuri

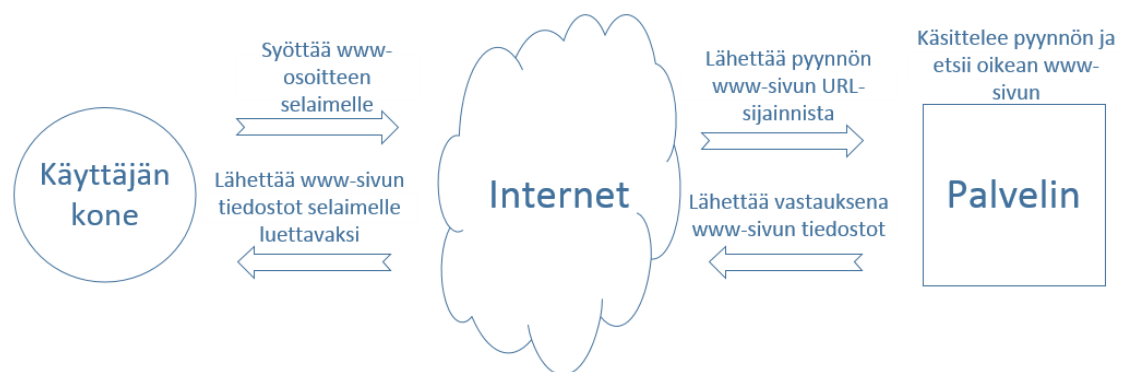
Ulkoinen www-arkkitehtuuri on suunniteltu käyttäjäystävälliseksi siten, että kuka tahansa pystyy käyttämään ja selaamaan www-sivuja. Käyttäjälähtöisyys on jälleen kerran avainsana: sivustojen pitää kyetä tarjoamaan käyttäjilleen juuri sitä, mitä hän on hakemassa, ja kaiken pitää toimia loogisesti (Nielsen, J. 2012).

Käyttäjän kannalta tärkein elementti Internetin selaamiseen tietokoneen ja toimivan Internet-yhteyden lisäksi on tietokoneelle asennettu www-selain. Tämä ohjelma toimii ensisijaisesti graafisena käyttöliittymänä, jonka avulla käyttäjä pystyy selaamaan HTML-/PHP-dokumentteja (eli www-sivuja) tai muita Internetiin ladattuja tiedostoja ja tarkastelemaan niitä. Selain tulkitsee dokumenttiin kirjattuja arvoja ja ”pukee” ne luettavaan muotoon. Ilman modernia selainta ainoa asia, joka käyttäjälle välittyisi, olisivat pitkät litaniat koodia, jota kovin harva osaisi tulkita oikein. Sivuston ulkoasuun vaikuttavat HTML- ja CSS-tiedostoissa määritellyt arvot, joista lisää kappaleessa 3.3.1. (Robbins 2012, 23)

Kaikki Internetissä kulkeva data tallennetaan palvelimille. Palvelimet ovat tietokoneita, joiden kovalevyille www-sivustojen informaatio ja muut Internetiin yhteydessä olevat tiedostot on tallennettu ja jotka ovat yhteydessä verkkoon muiden koneiden tavoittamiseksi. Palvelin voi olla yhden organisaation verkkodatan varastoimiseen pyhitetty kone tai kuulua palvelintilaa yksityisille tahoille myyville yritykselle (Robbins 2012, 21).

Www-sivun avaaminen selaimella tapahtuu seuraavalla tavalla: avattuaan selaimen käyttäjä kirjoittaa osoiteriville haluamansa sivuston www-osoitteen. Selain hakee sille syötettyjen arvojen pohjalta www-sivun URL-arvon (Uniform Resource Finder) eli sijainnin palvelimelta. Palvelin tulkitsee selaimen lähettämän pyynnön ja palauttaa

vastineeksi HTML- tai vastaavan tiedoston joka muodostaa www-sivun. Näitä dokumentteja käyttäjä voi sitten tarkkailla (Ks. Kuvio 1.). Tämä on perusidealtaan koko Internetin toiminnan ydin: yhteyden muodostaminen koneelta/palvelimelta toiseen asiakas/palvelin-periaatteella, yhteyden toimiessa koneiden välillä sen ajan, joka tiedonsiirtoon tarvitaan. Kaikki koneet, jotka ovat yhteydessä Internetiin, toimivat verkossa tasavertaisessa asemassa toisiinsa nähden (Robbins 2012, 30).



Kuvio 1. Internetin toimintaperiaate

3.3 HTML

HTML (Hypertext Markup Language) on www-sivun kehitystyön perusta, merkintäkieli, jonka pohjalta kaikki graafiset, Internet-selaimella avattavat www-sivustot on rakennettu. Se on lyhyesti sanottuna koko World Wide Webin julkaisukieli, ja tärkein asia, jota www-kehitystä opiskeleva tulee työssään tarvitsemaan (Robbins 2012, 12). HTML:n tiedostopääte on muotoa "file.html" (Robbins 2012, 50).

HTML-koodilla määritellään web-sivun rakenne ja sisältö. Kaikki sivulla näkyvä teksti kirjoitetaan HTML-koodilla ja samalla HTML sisältää toimintoja/arvoja jotka määrittelevät mitä erilaisia osioita sivulla esitetään ja miten kukin graafinen elementti asetuu. Näitä arvoja kutsutaan *tageiksi* (Robbins 2012, 26). Myös kuvien hakeminen palvelimelta sivulle toteutetaan HTML-kieltä käyttäen (Robbins 2012, 28).

HTML-dokumentti on lähtökohtaisesti muotiltaan täysin samanlainen, käytettiin sitä missä tahansa kontekstissa. Se koostuu samoista pääelementeistä, joita kaikki www-sivut käyttävät; sivun varsinaisen sisältöosuuden määrittelevässä rungossa on merkittävimmät rakenteelliset erot (Schafer 2010, 18).

```
1 <html>
2
3   <head>
4
5
6   </head>
7
8   <body>
9
10
11
12   </body>
13
14 </html>
```

Kuvio 2. HTML:n pohjarakenne

```

1  <!DOCTYPE html>
2
3  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4    <head>
5
6      <link rel="stylesheet" type="text/css" href="css/style.css"
          media="screen" />
7      <title> A simple web page </title>
8
9      <meta name="description" content="A basic example web site" />
10     <meta name="keywords" content="homepage, basic, web site, practice" />
11     <meta name="author" content="The Creator" />
12     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
13
14   </head>
15
16   <body>
17
18     <div id="container">
19
20       <h1>Welcome to our web site!</h1>
21
22       <p> Lorem ipsum dolor sit amet, consectetur adipiscing
          elit. Etiam at ullamcorper turpis. Sed quis justo pharetra,
          pretium est eu, rhoncus magna. Praesent posuere mollis
          fringilla..</p>
23
24     </div>
25
26   </body>
27 </html>

```

Kuvio 3. Sama HTML-dokumentti laajennettuna

HTML-versioita on luotu useita vuonna 1991 luodusta HTML1:stä nykyisin käytetyimpiin HTML4.01 ja XHTML:ään sekä paraikaa kehitteillä olevaan HTML5:een. HTML on jatkuvasti kehittyvä ohjelmointikieli, johon lisätään käytössä ilmenneiden tarpeiden ja ongelmien mukaan uusia toimintoja ja korjataan vanhoja. (Raggett, D., Lam, J., Alexander, I.F. & Kmiec, M. 1998.) Etenkin HTML5 on aiempiin versioihin nähden merkittävä loikka www:n kehityksen kannalta, mutta tästä kerrotaan tarkemmin luvussa 3.3.2.

3.3.1 Tagit

Tageiksi eli tunnisteiksi kutsutaan HTML-koodissa käytettäviä dokumentin osioiden määrittelyä varten tarkoitettuja arvoja. Niiden tarkoitus on auttaa erottelemaan eri-

lainen sisältö toisistaan. Jotta erilaiset sivulla näkyvät elementit, kuten teksti ja kuvat, voidaan esittää selaimella, ne täytyy kirjata asiaankuuluvien tagien sisään. HTML-dokumentti ei nimittäin osaa tulkita suoraan kirjoitettua tekstiä ja komentoja sellaisinaan. Tagit on luotu tätä varten; jokaisella tagilla on ennalta määritetty tarkoitus, joka vaikuttaa siihen, miten HTML:ään kirjoitettu sisältö näkyy tai mitä toimintoja se tekee, kun sitä luetaan www-selaimella (Schafer 2010, 7).

Erilaisia tagien kohteita ovat esimerkiksi

- sivun eri elementit (yläosa, runko-osa, loppuosa) ja niiden sijoittelu (yhdessä CSS:n kanssa)
- sivustolla näkyvät kuvat
- www-sivujen linkit
- erilaiset sivuston tekstielementit, esim. otsikot/runkoteksti.

Tagit toimivat seuraavalla tavalla: tagilla on sekä alku- että lopputagi, joiden väliin kirjataan vaadittava arvo, esim. sivulle lisättävän kuvan tapauksessa kuvan sijainti palvelimella. HTML osaa siten tulkita koodin perusteella annetut komennot, minkä johdosta määritelty kuva tulostuu selaimelle (Robbins 2012, 55).

<taginimi>sisältö</taginimi>

Tageilla on myös erittäin suuri merkitys CSS:n käytössä. Tästä kerrotaan tarkemmin luvussa 3.4.

3.3.2 HTML5

HTML on edennyt historiansa aikana viidenteen versioonsa. HTML5 tarjoaa lukuisia muutoksia aiempiin standardeihin verrattuna, mukaan lukien (HTML Tutorial 2014):

- lukuisia uusia elementtejä sekä tageja
- CSS3-tuki

- video- ja äänitiedostojen sisällyttäminen HTML-tiedostoon (aiemmin tätä varten tarvittiin esim. erillisen Flash-liitännäisen käyttöä)
- 3D-grafiikan esittämiseen vaadittava tekniikka
- paikallisen tallennustilan tuki
- paikallinen SQL-tietokanta (ei tarvitse luoda erillistä tietokantaa)
- web-sovellusten tuki

HTML5:n selkeä edistysaskel Internetin yleisen kehityksen kannalta on sen suunniteltu kyky mahdollistaa multimedian sisällyttäminen www-sivuun entistä vaivattomammin. Video ja ääni tulevat olemaan yhä merkittävämpiä elementtejä interaktiivisen www-sivun kehityksessä ja käytössä. Siinä missä aiemmin www-sivu on vain kyennyt noutamaan näitä ohjelmia ja ominaisuuksia ulkoisesta lähteestä, HTML5 kykenee suorittamaan nämä toiminnot itsenäisesti (Robbins 2012, 181).

HTML5 tulee korvaamaan kaikki aiemmat HTML:n versiot – HTML4 ja XHTML – ja tulee korjaamaan erinäisiä niissä ilmenneitä ongelmia. Kuitenkin tätä versiota ei ole vielä omaksuttu täydellisesti web-kehityksessä, ja valtaosa Internetin verkkosivuista on yhä luotu vanhempia standardeja mukailten (Parfeni 2013). HTML5:n käyttöönotto on kuitenkin ollut erittäin nopeasti kasvavaa varsinkin yritysmailmassa (Levine 2014).

3.4 CSS

CSS (Cascading Style Sheets) on WWW-sivun suunnittelussa käytettävä ohjelmointikieli, jonka avulla HTML-dokumentti voidaan esittää selaimella graafisesti halutulla tavalla (Robbins 2012, 207). CSS ja HTML kulkevatkin hyvin pitkälti käsi kädessä. CSS-dokumentin tiedostopäätte on muotoa "file.css" (Robbins 2012, 213).

CSS määrittää HTML-dokumentille kirjatulle tekstille ja muille elementeille erinäisiä tiedoston ulkoasuun liittyviä arvoja esim. määrittämällä näiden muotoilun ja asettelun. Nämä tyyliohjeet voivat vaikuttaa kerralla koko dokumentin ulkoasuun tai vain yksittäiseen osioon (Robbins 2012, 207).

Huomattavaa on, että käytännössä www-sivu ei tarvitse toimiakseen CSS-arvoja, sillä www-selain osaa tulkita HTML-koodin omien dokumentin näyttämiseen suunniteltujen oletusarvojensa avulla ja tulostaa yksinkertaisen sivun tarkasteltavaksi. Ilman CSS:ää sivu koostuisi kuitenkin vain mustasta tekstistä valkoisella taustalla elementtien asettuessa lineaarisesti toistensa perään ilman minkäänlaista graafista muotoilua. Aikoinaan, kun Internetiä käytettiin vain kirjallisten dokumenttien jakamiseen, tämä malli sopi varsin hyvin käyttötarkoitukseensa. Kuitenkin jo muutama rivi CSS-koodia voi antaa mille tahansa HTML-dokumentille lukijaa miellyttävän graafisen ilmeen (Robbins 2012, 207-208).

HTML-koodissa määritellyt tagit ovat ensiarvoisen tärkeä osa CSS:n toimintaa. CSS nimittäin antaa näille tunnisteille ne arvot, jotka määrittävät elementtien ulkoasun selaimella tarkasteltuna. Seuraavassa kuvia HTML- ja CSS-tiedostoista, joissa viitataan samoihin tageihin.

```
44  
45  
46 <h1>Welcome to our web site!</h1>  
47  
48 <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Etiam at ullamcorper turpis. Sed quis justo pharetra, pretium est  
eu, rhoncus magna. Praesent posuere mollis fringilla. Nullam  
condimentum, nibh non bibendum fermentum, elit est sollicitudin  
sem, sollicitudin porta risus ante et ligula. Sed imperdiet dolor  
eu porttitor consectetur. Etiam luctus quis nisl eget porta. Cras  
vel metus at metus volutpat faucibus. Aenean vel enim in velit  
ornare pulvinar. Maecenas lobortis sapien et dui feugiat volutpat.  
</p>
```

Kuvio 4. HTML-dokumentti. Huomaa h1- ja p-tagit

```
89
90 h1 {
91     font-size:2em;
92     font-style: italic;
93     clear: both;
94 }
95
96
97
98 p {font-size:120%;
99     margin: 20px 0;
100 }
101
```

Kuvio 5. CSS-dokumentti, joka antaa tyyliarvoja em. HTML-tiedostolle

CSS voidaan lisätä osaksi HTML:ää kahdella tavalla. HTML-dokumentin sisäistä tietoa käsittelevään osioon voidaan luoda oma osio CSS:lle, johon kirjataan sivun elementeille osoitetut tyyliarvot (Robbins 2012, 209–212). Tämä metodi sopii käytettäväksi vain äärimmäisen yksinkertaisen sivun tapauksessa, jolloin CSS-arvoja on hyvin vähän esillä. Käytännössä ei ole järkevää pyhittää kymmeniä rivejä HTML-dokumentista CSS:ää varten (mikä on monien www-sivustojen tapauksessa varsin normaali määrä), sillä se on ohjelmoijan kannalta raskas lukea ja muokata.

Toinen, paljon selkeämpi ratkaisu on luoda CSS:lle oma dokumentti, johon tarvittavia arvoja voidaan kirjata. HTML-dokumenttiin lisätään lyhyt viittaus CSS-tiedoston sijainnista, ja HTML osaa hakea tarvittavat arvot automaattisesti. Täten samaa tyyliarvojen sarjaa voidaan käyttää useissa HTML-dokumenteissa vain kirjoittamalla kuhunkin lyhyt koodinpätkä sen sijaan, että jokaiseen tulisi rivikaupalla tekstiä (Robbins 2012, 213).

3.5 Työvälineet

3.5.1 Tekstieditori

Kaikkein yksinkertaisimmillaan www-sivu voidaan suunnitella ja luoda normaalia tekstieditoria käyttäen (Notepad/Muistio). Kyseinen työkalu löytyy jokaisesta tietokoneesta käyttöjärjestelmästä riippumatta, ja sitä käytetään käytännössä kaikessa ohjelmointityöskentelyssä vähintään ohjelmapohjana (Schafer 2010, 309).

Tekstieditoria käytetään sekä HTML- että CSS-tiedostojen luomiseen. Koodi kirjoitetaan dokumenttiin ja tallennetaan joko muotoon "tiedostonimi.html" (Robbins 2012, 50) tai "tiedostonimi.css" (Robbins 2012, 213). Tällä tavalla ne muokkautuvat automaattisesti joko HTML- tai CSS-dokumentiksi. Tämä on ensiarvoisen tärkeää muistaa, sillä HTML-koodin tallentaminen pelkäksi tekstitiedostoksi (.txt) ei tee siitä HTML-tiedostoa, eikä selain täten pysty lukemaan sitä sellaisena.

Ohjelmoijien tarpeita ajatellen on luotu lukuisia ilmaisia tai maksullisia tekstieditoriohjelmiä, jotka tarjoavat erilaisia työkaluja koodin kirjoittamiseen ja tunnistamiseen. Avoimeen lähdekoodiin perustuva Notepad++ ja maksullisen lisenssin tarjoava Sublime Text ovat erittäin suosittuja esimerkkejä laajemman toimintoskaalaan tarjoavista teksti- ja lähdekoodieditoreista, joita esiintyy runsaasti sekä ammatti- että harrastekäytössä (Robbins 2012, 16).

3.5.2 Www-selain

Selain on ensiarvoisen tärkeä paitsi nettisurffaajalle, myös www-sivun suunnittelijalle. HTML- ja CSS-dokumenttien toimivuus täytyy nimittäin testata tulevassa käyttöympäristössään, jotta voidaan selvittää koodin toimivuus ja paikantaa mahdolliset virheet ennen sivuston julkaisua.

Www-sivujen testausta varten on suotavaa asentaa koneelle useampia selaimia ja testata sivuston koodin toimivuutta niillä kaikilla, sillä eri selaimilla voi olla suuriakin

eroja siinä, miten ne esittävät HTML:n ja CSS:n ulospäin (Duckett 2010, 599). Suositeltavat ja tämän hetken käytetyimmät selaimet, joilla testausta kannattaa tehdä, ovat Internet Explorer, Google Chrome, Mozilla Firefox, Safari (Applen Mac-tietokoneiden vakioselain) ja Opera (Robbins 2012, 17, 23). Jos halutaan olla erittäin perinpohjaisia, on järkevää asentaa näistä selaimista sekä uusimpia että joitakin vanhempia versioita, sillä kaikkihan eivät välttämättä omaa täysin samoja selaimia eivätkä päivitä selaimiaan uusimpaan versioon.

3.5.3 FTP-palvelinohjelmat

Monet palvelintilaa vuokraavat yritykset tarjoavat omasta takaa (yleensä selaimella avattavan) ohjelman, jolla www-sivuston omistaja pääsee käsiksi palvelimeen ja pystyy siten muokkaamaan oman palvelintilansa sisältöä esim. siirtämällä sinne HTML-dokumentteja ja muita tiedostoja. Yksinkertaisuudestaan huolimatta tämä on varsin kömpelö ja aikaa vievä tapa sivuston nopeaa muokkausta silmällä pitäen, ja siksi on kehitelty erilaisia FTP-formaattia (File Transfer Protocol eli kahden koneen välisen tiedostojen siirron mahdollistava tekniikka) tukevia ohjelmia. Tällaisen ohjelman avulla www-ohjelmoija voi nopeasti saada suoran yhteyden palvelinkoneeseen ja siellä sijaitseviin omiin tiedostoihin ja lähettää muokatun HTML-tiedoston eteenpäin (Schafer 2010, 321).

Esimerkkejä käytetyistä FTP-palvelinohjelmista ovat maksullinen CuteFTP sekä avoimeen lähdekoodiin perustuva FileZilla (Schafer 2010, 325).

3.5.4 Kuvankäsittelyohjelmat

Web-sivun graafisessa suunnittelussa hyödynnetään usein kuvankäsittelyohjelmia, kuten Adoben PhotoShop ja ilmaisena jaettava, avoimeen lähdekoodiin perustuva GIMP. Näillä voidaan luoda graafinen www-sivun layout eli "leiska", joka antaa kuvan siitä millaiselta web-sivun kuuluisi näyttää. Kuvankäsittelyohjelmalla voidaan myös

suunnitella kuvia/grafiikoita, joita käytetään valmiissa sivustossa. (Robbins 2012, 507)

Kuvankäsittelyohjelmat kuuluvat pääsääntöisesti graafikoiden vakiokalustoon, mutta koska www-sivujen suunnittelu vaatii vähintään jonkinlaista graafista silmää, on verkkosivun työstäjän hyvä opetella edes suuntaa antavan sivuston muotin suunnittelu ja luominen. Tämä saattaa hyvinkin nopeuttaa varsinaisen työn toteuttamista.

4 TUTKIMUKSEN TOTEUTTAMINEN

Opinnäytetyön lähtökohtana oli luoda yksinkertainen WWW-sivu HTML:ää ja CSS:ää käyttäen ja kirjoittaa työn vaiheista manuaali, jonka tarkoituksena on toimia konkreettisenä, helppolukuisena oppaana www-sivun kehityksen toimenpiteistä Web Page Development -opintojakson opiskelijalle. Lukijaa opetetaan soveltamaan omia opintojaksolla opittuja taitoja käytännössä, eikä tuossa opintojen vaiheessa olekaan olennaista luoda massiivisia, runsaasti sisältöä omaavia sivustoja, vaan kyseessä ovat puhtaat perusasiat.

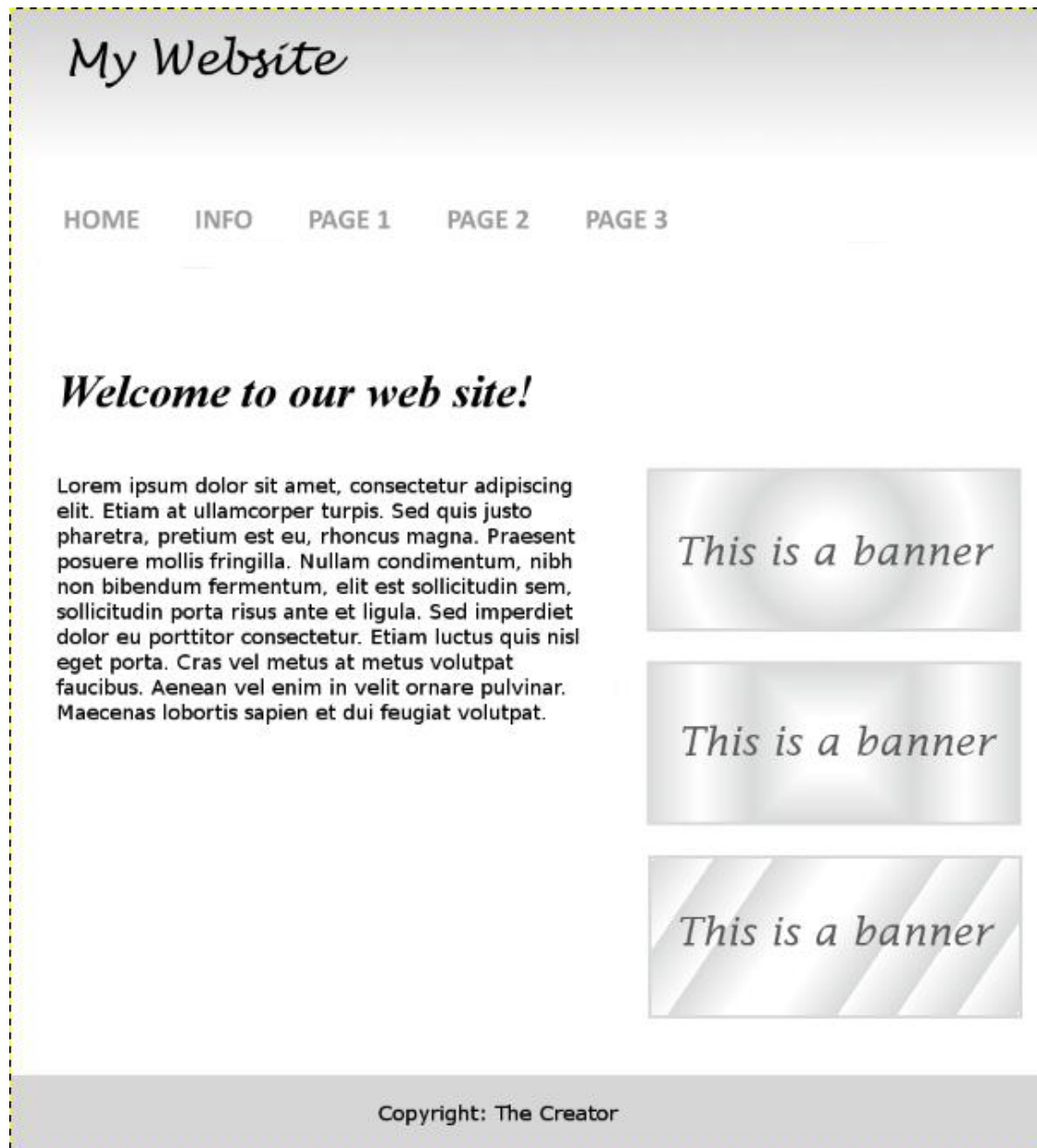
Työn vaiheet olivat seuraavat: www-sivun layoutin toteuttaminen kuvankäsittelyohjelmalla, sivun luominen HTML- ja CSS-koodia käyttäen ja edellä mainittujen toimenpiteiden kirjaaminen ylös manuaaliin.

4.1 Www-sivun layout

Työskentely aloitettiin www-sivun suunnittelusta. Tämä työvaihe valittiin osaksi manuaalin toteuttamista kahdesta syystä: ensinnäkin, jotta se helpottaisi valmiin sivuston toteuttamista, sillä visuaalista työtä tekeväille www-sivujen koodaajalle on paljon helpompaa ja nopeampaa työstää tehtäväksi annettua sivua, kun hänellä on selkeä

kuva siitä, miltä lopputuloksen kuuluu näyttää. Toisekseen manuaalin lukijan on hyvä oppia www-sivun suunnittelun ja luomisen kokonaisprosessi, joka alkaa sivun layoutin toteuttamisesta vaadittujen tarpeiden mukaisesti. Parhaassa tapauksessa lukija voi saada manuaalista virikkeen alkaa itse opettelemaan WWW-sivujen luonnostelua. Visuaalista silmää ja lukutaitoa on äärimmäisen tärkeitä opetella ja harjaannuttaa, sillä onhan Internetissä kyse yhdestä visuaalisen median muodosta.

Www-sivusta pyrittiin tekemään yksinkertainen ja helposti luettava, mutta samalla sellainen joka pitää sisällään standardin sivun olennaisimmat piirteet. Hyvin toteutetun www-sivun kuuluu olla toiminnaltaan looginen ja visuaaliselta ilmeeltään ihmisilmälle miellyttävä. Elementit on sijoitettava prioriteettien mukaan laskevassa järjestyksessä (ylimpänä sivun logo, sitten valikkopalkki, jonka jälkeen vasta alkaa varsinainen sisältö), ja värien ja muiden grafiikoiden pitää olla riittävän neutraaleja (harmaata sävyä valkoista taustaa vasten), jotta sivuston selaajan huomio ei kiinnity epäolennaisuuksiin. Lisäksi mitään ylimääräistä, huomiota herättävää sisältöä ei sivulla ole.



Kuvio 6. Sivun layout

Lukijan on tärkeä oppia havaitsemaan layoutista sivun eri elementit, jotta hän pystyy luomaan Www-sivun vaivattomammin. Hänelle haluttiin manuaalissa osoittaa layoutista nämä elementit, jotta varsinaisen koodaamisvaiheen aikana ei tarvitsisi kuluttaa aikaa ja vaivaa niiden seikkojen määrittelymiseen.

Layoutin suunnittelun yhteydessä syntyy usein myös valmista grafiikkaa, joita voidaan käyttää sivuston toteuttamisessa. Lukijalle osoitettiin tästä layoutista kyseiset graafiset elementit.

4.2 Siirtyminen koodiin

Layoutin käsittelyn jälkeen aloitettiin varsinaisen sivuston työstäminen. WPD-opintojaksolla pääpainopiste on HTML- ja CSS-koodauskielissä, joten www-sivu toteutettiin juuri näitä hyödyntäen.

Lukijalle annettiin manuaalissa seikkaperäiset ohjeet sivuston toteuttamisen koko prosessista alkaen www-sivun kansiorakenteen luomisesta. Dokumenttien oikeaoppinen organisointi on ensiarvoisen tärkeää, jotta sivustoon tarvittavat tiedostot löytyvät helposti esim. jatkokehitystä silmällä pitäen.

Varsinainen HTML-koodaaminen aloitettiin määrittelemällä HTML-standardi. Tätä nimenomaista sivua varten standardiksi valittiin XHTML 1.0 siksi, että se on vielä tällä hetkellä erittäin yleinen käytössä oleva HTML:n versio. HTML5 tulee yleistymään jatkuvasti etenkin uusien Web-sivujen keskuudessa, mutta manuaalin ja perusasioiden opetuksen kannalta oli olennaisempaa tutustuttaa lukija perinteiseen HTML-formaattiin.

HTML ja CSS seuraavat layoutin asettamia vaatimuksia. Sivuston luomisen suhteen ei annettu elementtien kokoihin jne. liittyviä ehdottoman tarkkoja ohjeistuksia, joita pitää seurata orjallisesti, vaan manuaalissa kehoitettiin kokeilemaan erilaisia asetuksia ja määritteitä. Kuitenkin www-sivun suunnittelussa ja toteuttamisessa painotetaan tietynlaista visuaalisen yhtäläisyyden säilyttämistä varsinkin silloin kun sivustoja suunnitellaan ja valmistetaan maksaville asiakkaille. Tähän viitaten lukijalle määriteltiin muutamia ehtoja, joiden täytyi toteutua sivun toteuttamisen yhteydessä.

HTML- ja CSS-tiedostoja työstettiin manuaalin mukaan niin sanotusti "ylhäältä alaspäin", eli dokumentin ylimmäinen elementti käsiteltiin ensimmäisenä ja siitä eteenpäin alenevassa järjestyksessä seurasivat muut sivun osiot. HTML-dokumenttiin kirjoitettiin www-sivun rakenteeseen vaikuttavat tiedot elementti kerrallaan, ja kunkin

osion valmistuttua siirryttiin CSS:n pariin muokkaamaan elementin graafista ulko-
 asua. Samaa kaavaa jatkettiin, kunnes www-sivu saatiin tehtyä valmiiksi.

```

20 <body> <!-- here starts body -->
21
22 <div id="header"> <!-- header div for site banner/other stuff -->
23  <!-- banner ie. the logo of the site -->
24
25
26 <div id="menu"> <!-- menu div for site navigation links -->
27 <a href="home.html" class="link">Home</a>
28 <a href="info.html" class="link">Info</a>
29 <a href="page1.html" class="link">Page 1</a>
30 <a href="page2.html" class="link">Page 2</a>
31 <a href="page3.html" class="link">Page 3</a>
32
33 </div> <!-- menu div ends here -->
34
35 </div> <!-- header div ends here -->

```

Kuvio 7. HTML-dokumentin valmiita elementtejä

```

12
13 #header {
14     width: 900px;
15     background-color: white;
16 }
17
18 #menu {
19     width: 900px;
20     margin-left: 30px;
21     background-color: white;
22 }
23

```

Kuvio 8. Em. HTML-dokumenttia täydennetään CSS:n avulla.

On yleisesti ottaen viisasta toteuttaa www-sivun luominen suunnitelmallisesti ja loogisesti, sillä siten koko prosessin pystyy hahmottamaan hyvin ja työ edistyy nopeasti. Kaikkiin www-osaajiin tämä ei tietenkään päde, ja monet tekevät verkkosivuja vaivatta hyvinkin vaihtelevin metodein, mutta vasta-alkajaa on hyvä ohjeistaa tekemään työ valmiiksi vaihe kerrallaan. Verkkosivu on helppo jakaa osiin ja sille on helppo määritellä selkeä rakenne alku- ja loppupisteineen, joten sen suunnitelmallinen toteuttaminen on vaivatonta omaksua.

4.3 Manuaalin kirjoittaminen

Www-sivun valmistuttua kaikki työn vaiheet layoutista HTML- ja CSS-koodiin kirjoitettiin manuaaliin. Layout esiteltiin dokumentin alussa, ja sen merkitys muutoin hyvin teknisessä www-sivustojen toteuttamisessa selitettiin lyhyesti ja tarkoituksenmukaisesti.

Jokainen tehty toimenpide kuvailtiin lyhyesti ja selkeästi ilman pitkiä teknisiä selityksiä. Lähtökohtana oli, että lukija hallitsee valmiiksi HTML:n ja CSS:n perusteet, joten näihin liittyviä termejä ei ollut tarpeellista selittää yksityiskohtaisesti, ja lisäksi liian suuret määrät tekstiä saattaisivat lannistaa kärsimättömämmän lukijan /vastaalkajan mielenkiinnon aiheeseen. Tästä syystä jokainen työn vaihe esitettiin myös kuvia hyödyntäen. Näin lukija kykenee paremmin hahmottamaan prosessin kulun ja jokaisen koodiin tehdyn muokkauksen tuloksen.

Koko manuaali kirjoitettiin englanniksi, sen ollessa Web Page Development -opintojakson ohjauskieli. Manuaalin kirjoittamisessa piti ottaa huomioon lukijoiden vaihteleva kielitaito, mikä osaltaan korosti tarvetta tehdä dokumentin tekstistä mahdollisimman suppeaa ja kielen yleisilmeestä mahdollisimman selkeää.

Next let's focus on the top element of the site, the header. Within it we'll include the banner of the web site and following that, separated by its own div tags we have menu element with links to other sites. Notice that the links have their own class "link"...

```

20 <body> <!-- here starts body -->
21
22 <div id="header"> <!-- header div for site banner/other stuff -->
23  <!-- banner ie. the logo of the site -->
24
25
26 <div id="menu"> <!-- menu div for site navigation links -->
27 <a href="home.html" class="link">Home</a>
28 <a href="info.html" class="link">Info</a>
29 <a href="page1.html" class="link">Page 1</a>
30 <a href="page2.html" class="link">Page 2</a>
31 <a href="page3.html" class="link">Page 3</a>
32
33 </div> <!-- menu div ends here -->
34
35 </div> <!-- header div ends here -->

```

Include the appropriate attributes to the CSS file regarding the header and menu, nothing fancy here.

```

12
13 #header {
14     width: 900px;
15     background-color: white;
16 }
17
18 #menu {
19     width: 900px;
20     margin-left: 30px;
21     background-color: white;
22 }
23

```

The end result should look something like this.



Kuvio 9. Ote manuaalista

Valmis manuaali saate- ja loppusanoineen tallennettiin PDF-tiedostoksi ja luovutettiin WPD-opintojakson opetuskäyttöön

5 POHDINTA

Tutkimuskysymys ja samalla ennalta määritelty tavoite oli seuraava: millaisilla työvälineillä Web Page Development -opintojakson opiskelijoiden kokonaisnäkemysten kehittymistä voitaisiin tukea? Tätä kysymystä silmällä pitäen päädyttiin selkeän ja haastavan opintomateriaalin toteuttamiseen, joka toimisi samalla JAMKin pedagogisten tavoitteiden valossa työkaluna opiskelijoiden itsenäiseen työskentelyyn.

Tutkimuksen tuloksina saatiin kaksi merkittävää asiaa. Olennaisimpana konkreettisenä tuotoksena saatiin Web Page Development -opintojakson opiskelijoille tarkoitettu verkkosivun toteuttamiseen perehdyttävä manuaali, joka tulee osaltaan tukemaan kyseisen opintojakson muuta opetustyötä ja ennen kaikkea vahvistaa tavoiteltua pedagogiikkaa.

Toinen saavutettu tulos oli tutkimuksesta kirjoitettu raportti. Sen lisäksi, että tämä dokumentti antaa kuvan tutkimuksen taustoista ja käytetyistä toteutusmetodeista, se valottaa myös opiskelijoille suunnatun opintoaineiston tuottamisen kokonaisprosessia. Tämän tutkimuksen ja sen tuloksena syntyneen opintomateriaalin on tuottanut ammattikorkeakouluopiskelija, joka on ollut vastaavassa asemassa kuin manuaalin kohderyhmä ja on täten erityisen hyvässä asemassa jakamaan oppimaansa tietoa eteenpäin.

Vastaavanlaista opintomateriaalin tuottamista ammattikorkeakouluopiskelijalta toisille voitaisiin hyvinkin tuottaa jatkossakin. Kehittämistutkimuksen puitteissa voitaisiin helposti uudistaa päivitystä kaipaavia aineistoja tai tuottaa täysin uutta materiaalia – esimerkiksi manuaaleja.

Tämän tutkimuksen yhteydessä syntynyttä manuaalia voidaan myös kehittää eteenpäin. Kyseisessä manuaalissa käsiteltiin pääsääntöisesti XHTML:n ja CSS:n toimintaa ja www-sivun tuottamisen perusasioita. Ottaen huomioon www-suunnittelun aiheiden laajan kirjon tämän dokumentin pohjalta voidaan luoda vaihtelevia manuaaleja tai muita opintomateriaaleja. Esimerkiksi HTML5:n pohjalle suunnitellun www-sivun

toteuttaminen ja tätä standardia hyödyntävien sovellusten luominen www-sivun yhteyteen olisi luonnollinen jatke tälle dokumentille.

Kokonaisuutena tutkimusta voidaan pitää onnistuneena, muutamaa takaiskua lukuun ottamatta. Tutkimuksen valmistuminen viivästyi alkuperäisestä aikataulusta (marraskuu 2013) lukuisten henkilökohtaisten esteiden vuoksi, ja toisaalta myös tekijän kokemattomuus kehittämistutkimuksen saralla vaikutti tähän. Tästä huolimatta tutkimus on saavuttanut tavoitteensa, ja jo käyttöön otettu manuaali on kerännyt rohkaisevan positiivista palautetta Web Page Development -opintojakson opiskelijoilta: heille osoitetun kyselyn perusteella manuaali on osoittautunut riittävän selkeäksi ja sisällöltään sopivan haastavaksi. Tämän pohjalta on hyvä jatkaa opintomateriaalin kehittämistä.

LÄHTEET

Bisneksen ja tietotekniikan osaaja. N.d. Tietojenkäsittelyn koulutusohjelma. Jyväskylän ammattikorkeakoulun kotisivut. Viitattu 8.11.2014.

<http://www.jamk.fi/fi/koulutus/liiketalouden-ala/tradenomi-tietojenkäsittely/>

Duckett, J. 2010. Beginning HTML, XHTML, CSS and JavaScript. Indianapolis (IN): Wiley Publishing, Inc.

HTML Tutorial. N.d. W3Schools Online Web Tutorials. Viitattu 8.11.2014

<http://www.w3schools.com/html/default.asp>

HTML5: what is it? 2011. Techradar-sivuston artikkeli. Viitattu 12.10.2014.

<http://www.techradar.com/news/internet/web/html5-what-is-it-1047393>

Kananen, J. 2012 Kehittämistutkimus opinnäytetyönä. Jyväskylän ammattikorkeakoulun julkaisuja 134.

Kehittämistutkimuksesta saadaan käytännöllistä tietoa opetuksen kehittämiseen.

2013. Verkkoartikkeli LUMA.fi:n sivuilla, julkaistu 9.4.2013. Viitattu 15.10.2014

<http://www.luma.fi/artikkelit/1913/kehittamistutkimuksesta-saadaan-kaytannollista-tietoa-opetuksen-kehittamiseen>

Levine, B. 2014. Report: HTML5 is the Future for Enterprise Developers. Verkkoartikkeli CMSWire.com:in sivuilla, julkaistu 12.2.2014. Viitattu 22.10.2014.

<http://www.cmswire.com/cms/customer-experience/report-html5-is-the-future-for-enterprise-developers-024137.php>

Nielsen, J. 2012. Usability 101: Introduction to Usability. Verkkoartikkeli Nielsen Norman Groupin sivuilla, julkaistu 4.1.2012. Viitattu 2.11.2014.

<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>

Parfeni, L. 2013. HTML5 Is Used by Almost 10 Percent of the Web, Google Says. Verkkoartikkeli Softpedia sivuilla, julkaistu 19.9.2013. Viitattu 22.10.2014.

<http://news.softpedia.com/news/HTML5-Is-Used-by-Almost-10-Percent-of-the-Web-Google-Says-384540.shtml>

Raggett, D., Lam, J., Alexander, I.F. & Kmiec, M. 1998. Raggett on HTML 4, Chapter 2.

<http://www.w3.org/People/Raggett/book4/ch02.html>

Robbins, J. N. 2012. Learning Web Design, 4th Edition. Sebastopol (CA): O'Reilly Media, Inc.

Schafer, S. M. 2010. HTML, XHTML, and CSS Bible, 5th Edition. Indianapolis (IN): Wiley Publishing, Inc.

Web Page Development. 2014. Opintojakson informaatio sivu.

https://asio.jamk.fi/pls/asio/asio_ects kuv1.kurssin_ks?ktun=HTO102W0&knro=&lan=e&ark=

LITTEET

Practical Tutorial for Web Page Development. Manuaali www-sivun toteuttamisen prosessiin JAMKin Web Page Development -opintojakson opiskelijoille.

Practical Tutorial for Web Page Development

**How to create a web page from scratch using a graphical layout as
model**

By Jussi-Pekka Karisaari

September 2014

Created for Web Page Development course at JAMK University of Applied Sciences

Author's welcoming words

Welcome to the Practical Tutorial for Web Page Development. This short document is mainly aimed for students at TIKO's Web Page Development course as a helpful guide to teach the basics of web design; however, anyone who wants to know more about putting web development skills into real use is more than welcome to use this as an introduction to the basics. This includes you, too!

Our purpose here is to learn basic web page development by creating an HTML page from scratch by using a graphical layout as a guide. I will not demand you to repeat the process I'm going through and thus create an exact replica of the page, instead I'd like this to be a simple guide that will hopefully give you valuable tips and insight in how to develop a page in general. Although if you like some challenge – and you possess some graphical skills of your own – you can work your way to make a web site of your own with these pieces of advice in mind.

I would also like to point out that there will be plenty of pictures instead of some technical mumbo-jumbo, so don't be alarmed if this document appears long. I also won't explain every little detail of web page development or describe every single HTML tag thoroughly, I expect you to have at least some basic skills in HTML and CSS coding.

If you're still interested in the subject after reading this brief introduction, then let's start shall we? I hope you enjoy our journey into the Practical Web Page Development!

From student to students,

Your guide

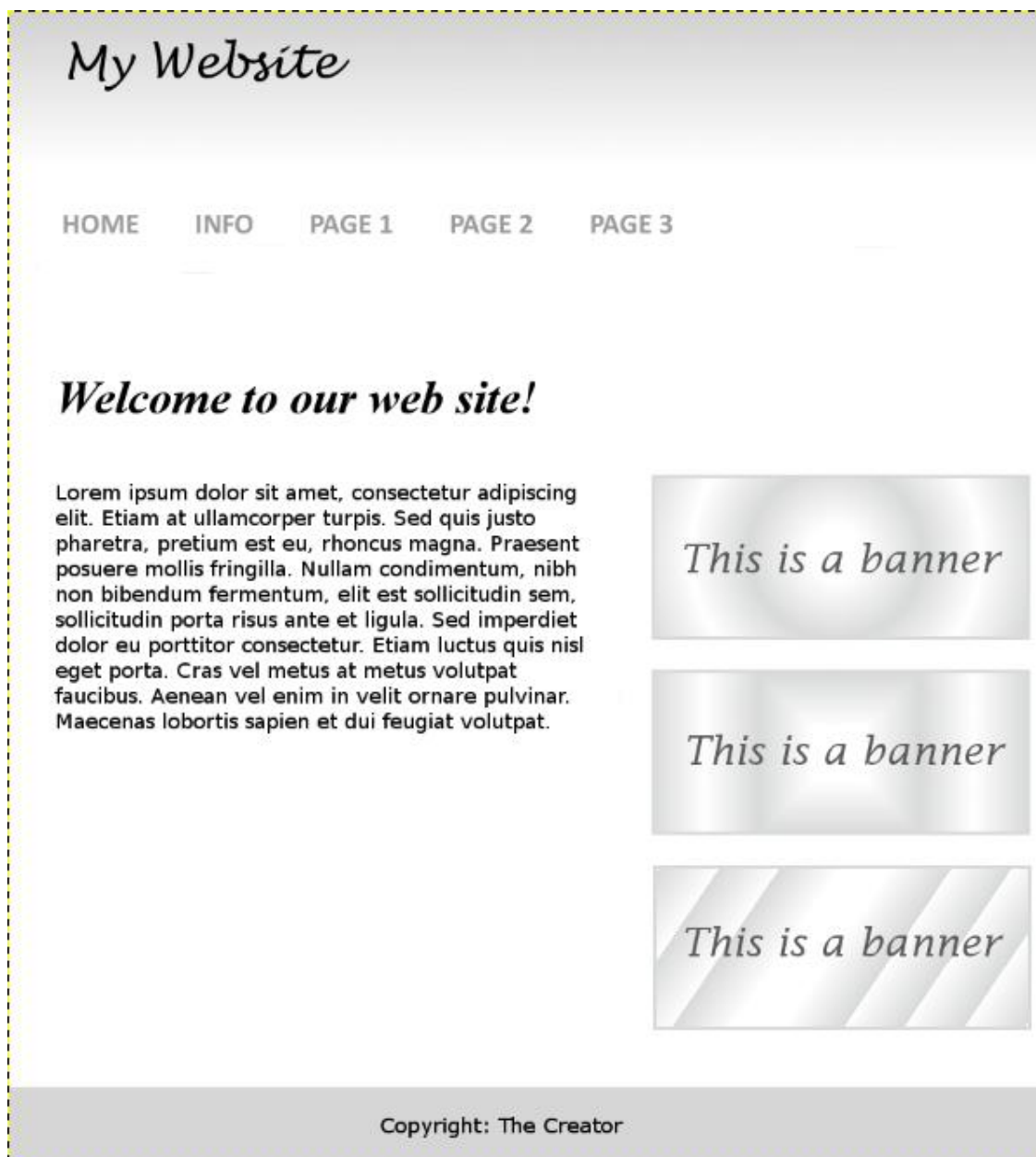
Jussi-Pekka Karisaari

E7332

HTI9S1

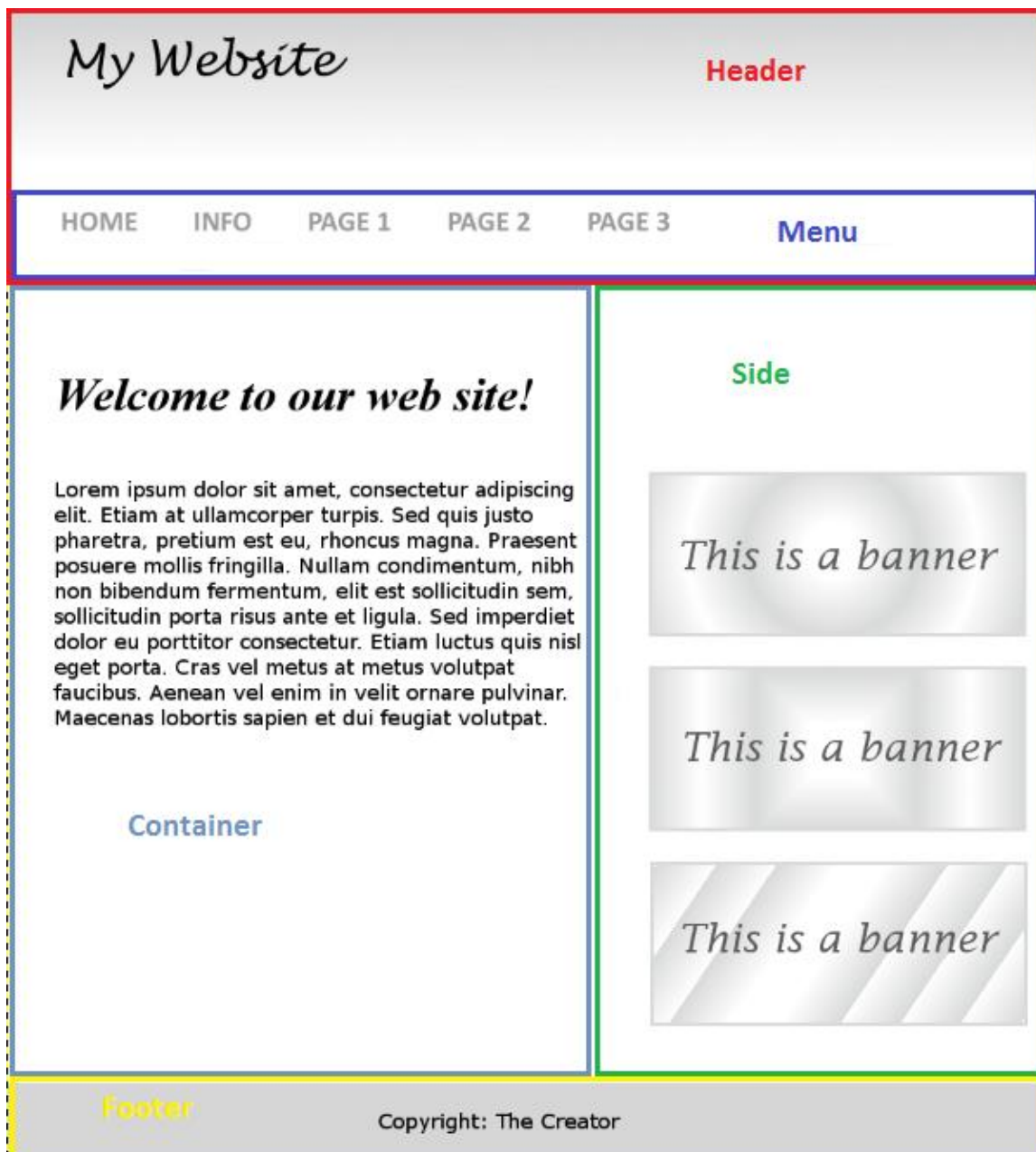
Tutorial

Starting with the very first thing, the web page itself. Or more specifically the model layout of the page we will work with. In most cases you, the web page developer will either have ready-made graphics at your disposal with which you can work on the web page or you may even work with a graphic designer who can and will help you in situations that demand graphical expertise. To demonstrate that we'll have a ready-made graphical layout. (Note: this has been created using [GIMP](#), a free graphics editing program. Not everything needs to be done with Photoshop after all.)



Looks like a fairly simple and common design, something you might see anywhere on the Internet. Also we do not need to create anything too complex after all, considering we are going through merely basics of web design.

In order to get a better idea of how to eventually start work on this web page we could split this layout – or HTML body – into sections or “elements” according a basic HTML structure. That way we’ll have a better understanding of the relations between different elements. If you’re used to working with HTML you may have already figured out these details by yourself, but nevertheless, we will have an image ready here.



When a graphics designer creates a layout for a web page the graphics that will be used as part of the actual product are usually conceived at the same time. We will have the separate files at our disposal, but let's spot them on our image while we're at it.

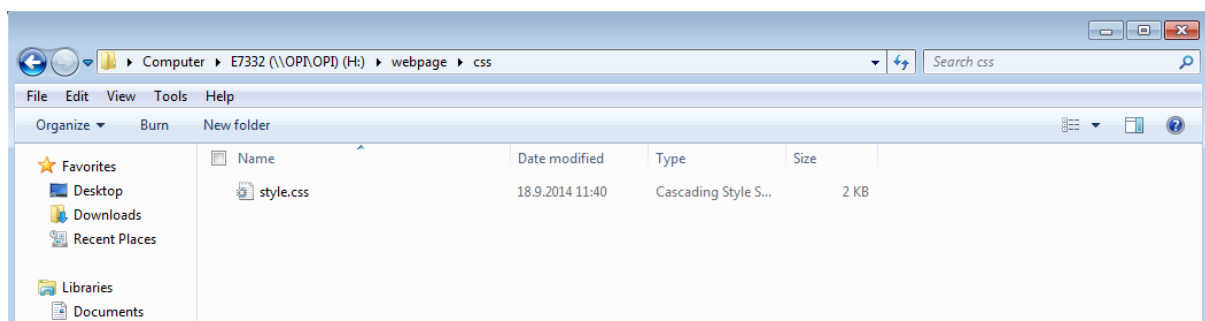
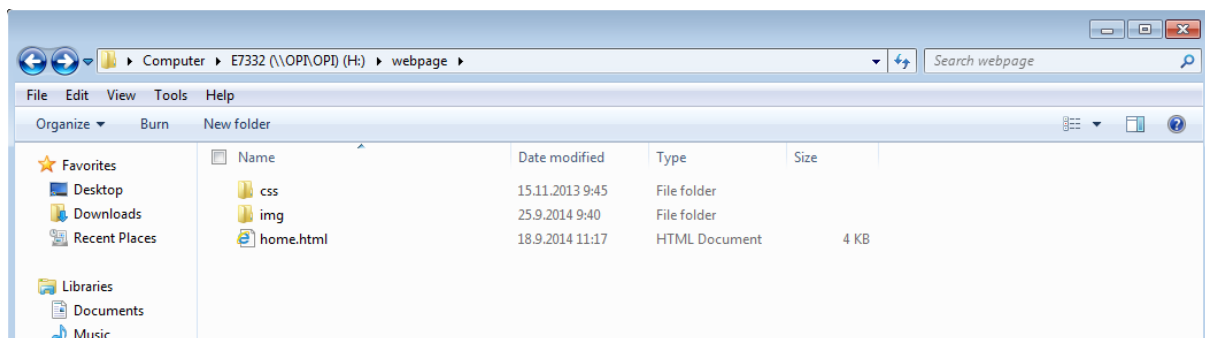


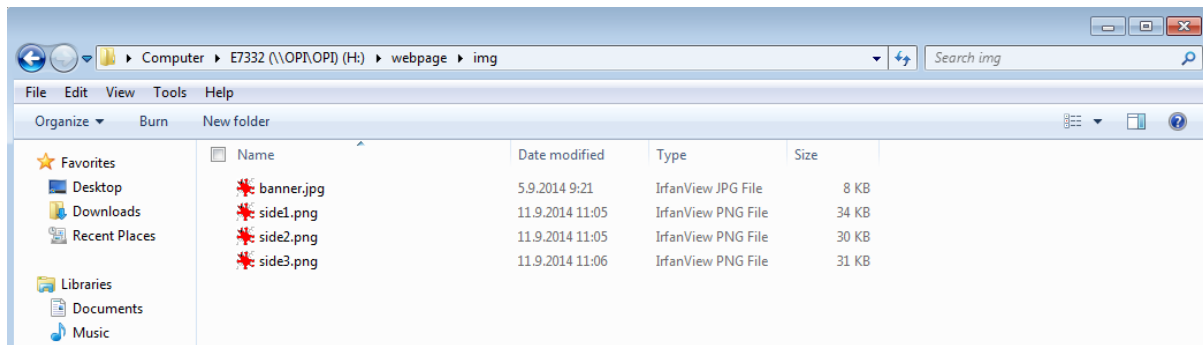
Pay heed that the final product should resemble the layout BUT by no means is there any set rule that orders you to make the web site 100% exact copy. Slight tweaking should be allowed as long as certain conditions are met, which are as follows:

- the width of the site = 900px
- main color schemes are **white, black & the grey hue #9b9b9b**
- all text should be aligned with the “My Website” logo
- side banners should be aligned with the main body of text (not with the “Welcome!” header text)

Now that we have gone through most of the layout, we can focus on actual web page development, starting with HTML. You can always check the layout as a reference as you work with coding.

Create a basic web page folder structure with HTML and CSS documents and a separate folder for images. To keep things simple and whatnot, let’s just name our root folder **webpage** and give equally simple, yet descriptive names for other files.





Let's make the HTML document in XHTML 1.0 format. Naturally, HTML5 would perhaps be a more appropriate option considering it's the latest phase in the evolution of web languages, however at the moment XHTML is still widely used, especially with older web sites.

Also add some appropriate meta-information for the web site. Remember the CSS document path.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7
8 <link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />
9 <title> A simple web page </title>
10
11 <meta name="description" content="A basic example web site" />
12 <meta name="keywords" content="homepage, basic, web site, practice" />
13 <meta name="author" content="The Creator" />
14 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
15
16
17 </head>
```

Now we can also start editing the CSS file. Naturally we'll begin with body element and thus define the web document's main structure.

```
1 body {
2     font-size:16px;
3     font-family:"Times New Roman",Georgia,Sans-Serif;
4     color:black;
5     text-align:center;
6     background-color: white;
7     margin: 0 auto;
8     width: 900px;
9 }
10
```

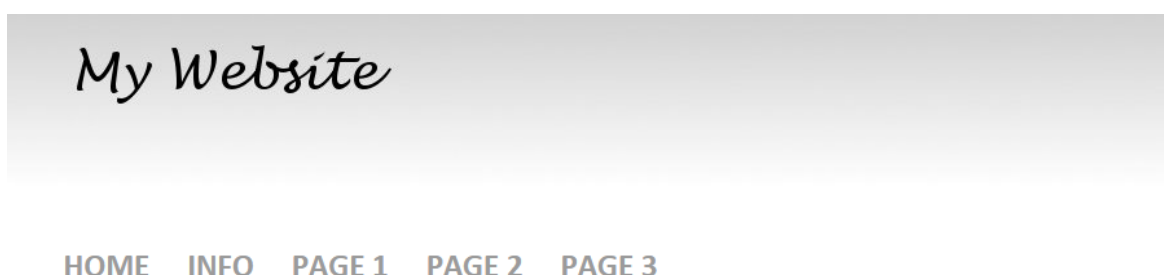
Next let's focus on the top element of the site, the header. Within it we'll include the banner of the web site and following that, separated by its own div tags we have a menu element with links to other sites. Notice that the links have their own class "link"...

```
20 <body> <!-- here starts body -->
21
22 <div id="header"> <!-- header div for site banner/other stuff -->
23  <!-- banner ie. the logo of the site -->
24
25
26 <div id="menu"> <!-- menu div for site navigation links -->
27 <a href="home.html" class="link">Home</a>
28 <a href="info.html" class="link">Info</a>
29 <a href="page1.html" class="link">Page 1</a>
30 <a href="page2.html" class="link">Page 2</a>
31 <a href="page3.html" class="link">Page 3</a>
32
33 </div> <!-- menu div ends here -->
34
35 </div> <!-- header div ends here -->
```

Include the appropriate attributes to the CSS file regarding the header and menu, nothing fancy here.

```
12
13 #header {
14     width: 900px;
15     background-color: white;
16 }
17
18 #menu {
19     width: 900px;
20     margin-left: 30px;
21     background-color: white;
22 }
23
```

The end result should look something like this.



Now we'll give the menu links (which we already defined with a separate class attribute) some flavor with little tweaks just for the sake of it. In addition to some padding (to make some room between links) and other cosmetic tweaks, we will set it so that the text color changes whenever a cursor hovers over it. Otherwise it'll retain its original color (note: the grey hue **#9b9b9b**).

```
48 .link {
49     float:left;
50     font-weight:bold;
51     font-size: 25px;
52     color:#9b9b9b;
53     text-align:left;
54     padding:0 14px 16px 15px;
55     text-decoration:none;
56     text-transform:uppercase;
57     font-family: "Calibri", Calibri;
58 }
59
60 .link:link      {color:#9b9b9b;}
61 .link:visited  {color:#9b9b9b;}
62 .link:hover    {color:black;}
63 .link:active   {color:black;}
64
```

This is how the above changes will affect the links when having a mouse cursor hovering over one of them.



Next we'll focus on the container div. Just include some irrelevant text, and remember to include a separate **h1 tag**.

```

40 <div id="container"> <!-- container div for main content of the site -->
41
42
43 <h1>Welcome to our web site!</h1>
44
45 <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam at ullamcorper turpis. Sed quis
justo pharetra, pretium est eu, rhoncus magna. Praesent posuere mollis fringilla. Nullam
condimentum, nibh non bibendum fermentum, elit est sollicitudin sem, sollicitudin porta risus ante
et ligula. Sed imperdiet dolor eu porttitor consectetur. Etiam luctus quis nisl eget porta. Cras vel
metus at metus volutpat faucibus. Aenean vel enim in velit ornare pulvinar. Maecenas lobortis sapien
et dui feugiat volutpat.</p>
46
47 <p>Nam lacinia est eget justo tristique, ultrices mollis odio vestibulum. Nulla ultricies dolor sed
elementum tempus. Donec leo est, luctus ut orci vitae, pretium pharetra mauris. Nulla facilisi. Mauris
et turpis elementum, porttitor lectus in, tincidunt libero. Morbi feugiat dolor vitae dictum dictum.
Pellentesque suscipit tortor ut fermentum mollis. Donec aliquam ac ante et hendrerit. Maecenas convallis
condimentum nunc vitae congue.</p>
48
49 <p>Proin eleifend metus nec ante tristique, pulvinar tincidunt est mollis. Pellentesque condimentum quam
at elit pellentesque, in hendrerit orci sodales. Pellentesque et tincidunt neque. Nullam pharetra diam
sed tincidunt luctus. Phasellus egestas eu sem id viverra. Mauris adipiscing luctus euismod. Cras
dapibus vulputate iaculis. Vivamus odio est, egestas pharetra commodo quis, ultricies at tellus. Aenean
sed scelerisque purus. Morbi metus mi, tincidunt nec risus et, viverra mollis nisi. Vestibulum lobortis
iaculis ligula, a feugiat dolor lacinia sed.</p>
50
51
52 </div> <!-- Here ends the container div -->

```

In CSS we'll define certain attributes for the container div and the `<h1>` and `<p>` tags. This is the first instance where we have to limit the width of an element, considering that we have to leave some room for the side div that follows soon after. The overall width of the document is 900px, so let's make the total width of the container 560px (since the side div will include at least 300px wide banners we need to leave appropriate amount of empty space); this'll include margin and padding values too. Remember that the text elements must be aligned with the banner logo, thus we'll add some extra margin values. This'll take some tweaking to find a perfect value.

```

22 #container{
23     text-align:left;
24     padding: 20px;
25     width: 495px;
26     float: left;
27     margin-left: 25px;
28 }
29

```

```
71  h1      {
72      font-size:2em;
73      font-style: italic;
74      clear: both;
75      }
76
77
78  p      {
79      font-size:120%;
80      margin: 20px 0;
81      }
```

This is how our site looks thus far.

My Website

[HOME](#) [INFO](#) [PAGE 1](#) [PAGE 2](#) [PAGE 3](#)

Welcome to our web site!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam at ullamcorper turpis. Sed quis justo pharetra, pretium est eu, rhoncus magna. Praesent posuere mollis fringilla. Nullam condimentum, nibh non bibendum fermentum, elit est sollicitudin sem, sollicitudin porta risus ante et ligula. Sed imperdiet dolor eu porttitor consectetur. Etiam luctus quis nisi eget porta. Cras vel metus at metus volutpat faucibus. Aenean vel enim in velit omare pulvinar. Maecenas lobortis sapien et dui feugiat volutpat.

Nam lacinia est eget justo tristique, ultrices mollis odio vestibulum. Nulla ultricies dolor sed elementum tempus. Donec leo est, luctus ut orci vitae, pretium pharetra mauris. Nulla facilisi. Mauris et turpis elementum, porttitor lectus in, tincidunt libero. Morbi feugiat dolor vitae dictum dictum. Pellentesque suscipit tortor ut fermentum mollis. Donec aliquam ac ante et hendrerit. Maecenas convallis condimentum nunc vitae congue.

Proin eleifend metus nec ante tristique, pulvinar tincidunt est mollis. Pellentesque condimentum quam at elit pellentesque, in hendrerit orci sodales. Pellentesque et tincidunt neque. Nullam pharetra diam sed tincidunt luctus. Phasellus egestas eu sem id viverra. Mauris adipiscing luctus euismod. Cras dapibus vulputate iaculis. Vivamus odio est, egestas pharetra commodo quis, ultricies at tellus. Aenean sed scelerisque purus. Morbi metus mi, tincidunt nec risus et, viverra mollis nisi. Vestibulum lobortis iaculis ligula, a feugiat dolor lacinia sed.

Our next objective is the side div and the banners that are included within it. Each banner leads to some website. We could give the images additional title information so that when the cursor hovers over them it'll reveal a descriptive text.

```
55 <div id="side"> <!-- side div for banners etc. -->
56
57 <a href="http://www.notexistingwebsite.com" title="Adventure starts here!"> </a> <!-- banner -->
58 <a href="http://www.anotherimaginarywebsite.com" title="Shall we try again...?"> </a> <!-- banner -->
59 <a href="http://www.didyualreadyfigureitout.com" title="Third time's the charm! Right?"><img src=
"img/side3.png" alt="side banner 3" /> </a> <!-- banner -->
60
61 </div> <!-- Here ends the side div -->
```

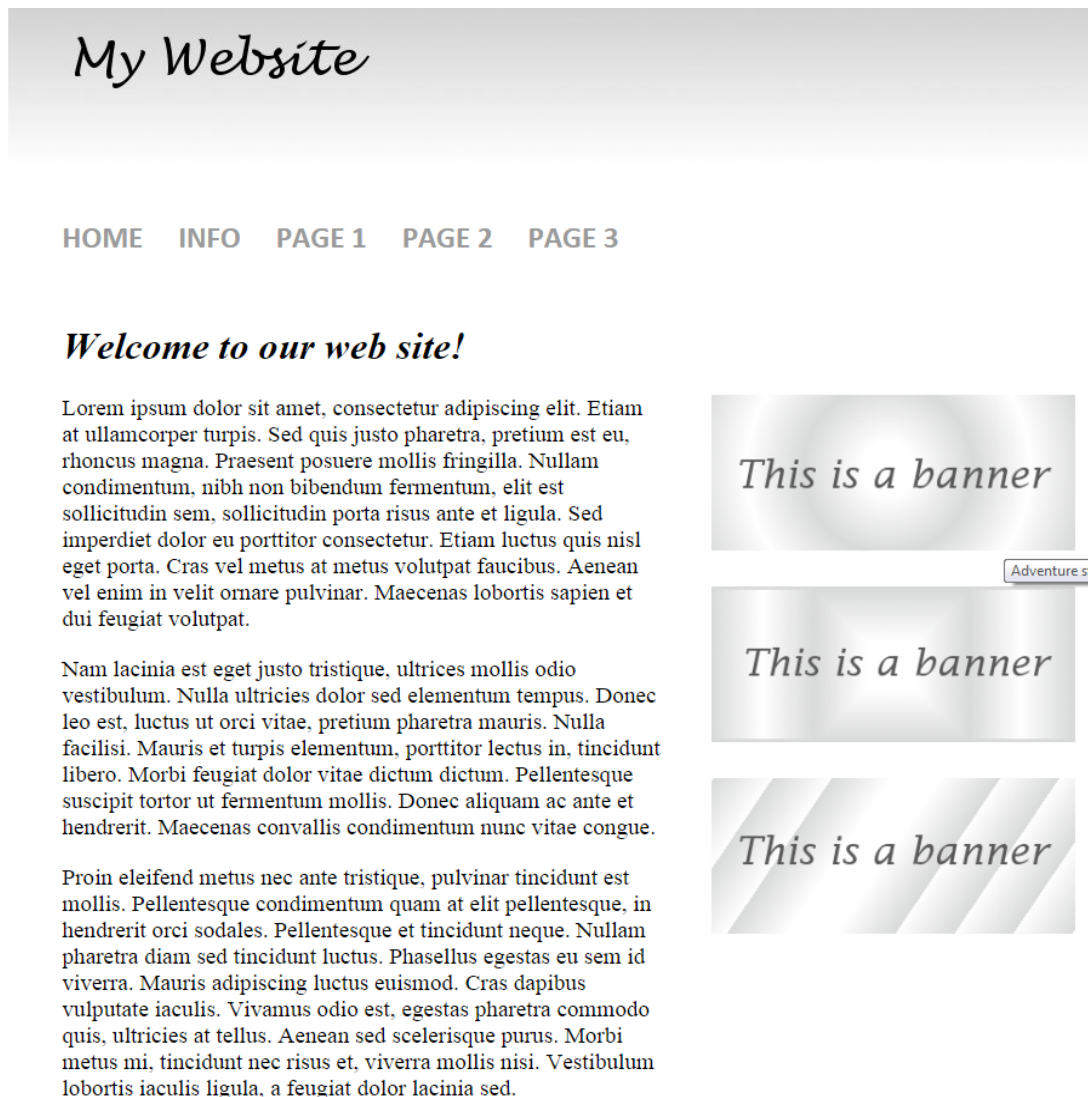
Back to the CSS. Remember the width of the side element: $900\text{px} - 560\text{px} = 340\text{px}$. And since the width of the banner images is 300px , some padding should fill the rest nicely. Also add some margin value to align the side div with the `<p>` text element, that'll make the site look more orderly.

After we've added some empty space between the banners I think we're done with the side element.

```
30 #side {
31     width: 300px;
32     margin-top: 100px;
33     padding: 0 20px 0 20px;
34     float: left;
35 }
```

```
76 img {
77     margin-bottom: 30px;
78     float: left;
79     background-color:white;
80 }
```

Like so.



The only thing we have left is to add the footer div and thus mark the end of the HTML document.

```
63 | <div id="footer">&#169; 2014 The Creator</div> <!-- Footer for copyright information -->
64 |
65 | </body> <!-- Here ends body -->
66 | </html>
```

Add some final touches to the CSS file!

```

37  #footer {
38      clear: both;
39      padding: 0.5em;
40      font-size: 120%;
41      color: black;
42      background-color: #9b9b9b;
43      text-align: center;
44  }

```

Voilà! We have ourselves a web site!



Aftermath

That wasn't such a hard task now was it? As you can see the cycle of web development from layout to actual web site is not a tremendously complex process if you know the basics. Naturally things might become different when you also have to create a layout from scratch, but in either case don't be alarmed. Designing a layout is not hard as the tools and software are readily available, some of them free to download and use (*cough* [GIMP](#) *cough*), and in fact if you are in any way talented in visual arts or even slightly interested in the subject you might learn to enjoy the graphical part of this job. Whether you're creating a web site strictly for your own personal use or working for a client with specific guidelines and objectives in mind my advice is the same: let your imagination flow freely! If you are willing to take risks and simply try out different solutions you might achieve great feats.

Now I'll leave the rest to you. I hope this tutorial has provided some helpful insight into the subject of Web Page Development and encouraged you to develop your own web projects. If you want you could focus on this very site here. Is there something you would change or add? How would you improve the visuals and usability of the web page? Are there certain visual differences when viewing the document with different browsers that should be fixed? Feel free to experiment with this page and practice on your own, for only through practice can you truly improve your skills in web development.

For my last words of sage wisdom (?) I'll say this: whenever a problem occurs when dealing with HTML, CSS or image editing, just use unashamedly the greatest tool a web developer has in his/her possession.

Google.