

Основи графічного дизайну та проектної графіки

Головатий Роман Русланович

Лекція 1

Тема: Основні поняття Web-дизайну

План заняття:

1. Вступ до предмету
2. Поняття web-дизайну

Вступ до предмету

Термін Web-дизайн складається з двох частин. Перша частина - Web - це скорочене написання слів World Wide Web - Всесвітня Павутина. Друга частина терміна - слово "дизайн" - походить від англійського слова design, що означає: проектувати, конструювати, планувати, креслити, створювати малюнок. Таким чином, попередній переклад терміна Web-design з англійської мови на українську мову може бути таким: "проектування для Інтернету".

Для уточнення сенсу цього терміну розглянемо існуючі поняття і визначення.

Дизайн в широкому сенсі слова - будь-яке проектування. Процес створення нових предметів, інструментів, обладнання. Дизайн у вузькому сенсі слова - це художнє конструювання. Його мета - проектування предметів, в яких форма відповідає їх призначенню.

Образотворчими засобами дизайну є: точка, лінія, фактура, текстура, колір, форма, пропорція і ін. Коротко розглянемо наступні терміни.

Текстура - характер поверхні будь-якого об'єкта, обумовлений його внутрішньою будовою, структурою, об'єктивними фізичними властивостями (дерево, метал, скло, тканина).

Фактура - зовнішні властивості поверхні об'єкта (гладка, шорстка, дзеркальна). Фактура характеризує якість обробки поверхні об'єкту.

Форма - зовнішній обрис, зовнішній вигляд, контур об'єкта.

Колір - властивість тіл викликати певне зорове відчуття у відповідності зі спектральним складом і інтенсивністю відбиваного або випускається ними видимого випромінювання. Основні параметри кольору - колірний тон, насиченість і світлин. Тон визначається спектральним складом світла, насиченість - кількістю додається сірого кольору, а світлина - яскравістю.

Поняття web- дизайну

Web-дизайн - новий жанр образотворчого мистецтва, заснований на застосуванні специфічних інструментів (комп'ютера та програм-редакторів), спеціальних мов програмування (HTML, JavaScript, Java, Ruby, Perl, C ++, PHP та ін.) І мережевих технологій (CGI, CSS, SSI та ін.). Метою Web-дизайну є проектування (створення) об'єктів (Web-сторінок, банерів, аплетів, скриптів, окремих елементів Web-сторінок), що розміщуються в Інтернеті.

Web-дизайнер (або україномовні терміни: Web-майстер, Веб-майстер) - художник-програміст, що займається Web-дизайном.

Крім обов'язків, пов'язаних з дизайном, Веб-майстру нерідко доводиться також виконувати обов'язки системного адміністратора і програміста (мати гарне уявлення про мережеві технології та мови програмування). Початківці Web-дизайнери розміщують на своїх домашніх сторінках власні літературні, графічні та музичні твори, результати наукових досліджень, фотографії, репродукції улюблених картин, кулінарні рецепти, основні відомості про себе, про свої захоплення. Це дозволяє їм відшукати в мережі однодумців, людей зі схожими інтересами, спілкуватися з цікавими людьми.

Більш кваліфіковані дизайнери створюють сайти, присвячені некомерційним організаціям: навчальним закладам, добровільним об'єднанням, політичним партіям, улюбленим спортивним командам і музичним групам.

Професійні Web-дизайнери розробляють комерційні проекти, що приносять прибуток: віртуальні магазини, пошукові системи, каталоги, інформаційні сайти. Професійно розроблений Web-сайт є одним з видів ефективної реклами продукції або послуг, зручним засобом для пошуку ділових клієнтів і партнерів, успішного проведення маркетингової політики.

Розробка серйозного комерційного проекту вимагає значних витрат часу, вкладення коштів і застосування кваліфікованої праці. Найчастіше складні Web-сайти розробляються колективно, так як доводиться вирішувати безліч складних технічних, інформаційних, психологічних та інших завдань.

Основним завданням професійних Web-дизайнерів є залучення великої кількості відвідувачів на створений ними сайт. І якщо на сторінки дизайнерів-аматорів заходять два-три відвідувача на тиждень, то комерційні сайти відвідують щодня тисячі людей. Цьому сприяє нерозривна єдність художнього оформлення сайту і його предметного змісту (єдність форми і змісту). Не останню роль у створенні популярності відіграє вміння "розкрутити" сайт.

Для створення комерційного сайту потрібні професійна майстерність (знання теорії, володіння навичками), вміння виділяти актуальні проблеми і хороший художній смак. При розробці сайту дизайнери використовують закономірності, накопичені в живописі, графіці, скульптурі, архітектурі, художньої фотографії, декоративно-прикладному мистецтві, графіті. Однак багато прийомів Web-дизайну є принципово новими. Тому часом говорять про Web-дизайн як про новий жанр мистецтва, який спирається на свої

образотворчі закони. Ці закони в даний час створюються нашими сучасниками і проходять практичну апробацію в мережі.

На відміну від класичних творів образотворчих мистецтв, в Web-дизайні не використовуються олівці, пензлі, мольберти, фарби. Тут потрібні інші інструментальні засоби (HTML-редактори, графічні, відео та звукові редактори, аніматори, редактори скриптів, спеціальні утиліти, програми для створення фотоальбомів, різноманітні мережеві технології, мови програмування, цифрові фотоапарати, сканери та ін.).

На Web-сторінках присутні не тільки текст і статичні зображення, але і анімація, відео, звуки. Зовнішній вигляд переглядаються Web-сторінок динамічно змінюється при виконанні користувачем деяких дій. Це робить Web-сторінку складним твором мистецтва, створюваним на стику літератури, журналістики, образотворчого мистецтва, кіно, телебачення, радіо, фотомистецтва. При цьому досвідчений професійний розробник добре уявляє контингент людей, частіше за інших відвідують його сайти, враховує психологію користувачів, підбирає теми, які найбільш популярні в мережі в даний момент часу. Переважна більшість Web-сторінок є продуктом мультимедіа.

Проектування сторінок або сайтів ускладнюється ще й тією обставиною, що одна і та ж сторінка відображається по-різному різними браузерами (Google Chrome, Netscape Navigator, Opera) і в вікнах різного розміру. У багатьох випадках браузер самостійно "вирішує" як буде виглядати завантажена сторінка. Змінити макет сторінки може і користувач, наприклад, шляхом відключення опції завантаження малюнків, анімації.

Для створення динамічних Web-сторінок (що володіють елементами штучного інтелекту) завантаження окремих елементів проводиться в залежності від часу доби або за допомогою генератора випадкових чисел.

Таким чином, змінюються програються музичні кліпи, малюнки, що завантажуються анекдоти і афоризми.

При проектуванні сайту дизайнерові доводиться звертати увагу на правильний вибір кольору шрифту і кольору фону. Рекомендується віддавати перевагу синім буквам на білому тлі, чорним на жовтому, зеленим на білому, чорним на білому, білим на синьому. При виборі кольорової палітри слід враховувати, що найбільшу чутливість очей має до жовто-зеленого кольору, а найменшу до фіолетового і червоного кольорів. Помічено, що різні кольори надають різний вплив на людину: червоний колір збуджує, а синій колір заспокоює.

Дані психологічних досліджень говорять про те, що людське око сприймає червоний, помаранчевий, жовтий кольори (і їх відтінки) як теплі. Синій, фіолетовий кольори і їх відтінки сприймаються людиною як холодні. Це пов'язано, мабуть, з тим, що сонце і вогонь містять червоні і жовті кольори, а сніг, лід - блакитні і сині кольори....

Як правило, теплі тони на малюнку мають властивість "наближатися", а холодні - "віддалятися". Іншими словами, предмети, намальовані теплими тонами, будуть здаватися розташованими ближче предметів, зображених холодними тонами.

Фахівцями художньої фотографії встановлено, що око глядача звертається насамперед до світлих тонів знімка і до тих місць, де є найвищий тональний контраст.

Білий, чорний і сірий кольори називаються ахроматичними, всі інші - хроматичними. Особливості впливу кольору на людину вивчає наука колористика.

Перелічимо деякі важливі рекомендації щодо створення Web-сайту. Доцільно не перевантажувати сторінку великим числом деталей. Це пов'язано з тим, що людина за короткий час здатний сприйняти і запам'ятати не більше 7-8 об'єктів.

Вважається, що чим менше розмір об'єкта, тим більше повинна бути його контрастність. Помічено, що чітко виражений фоновий малюнок на Web-сторінці підвищує стомлюваність очей і знижує ефективність сприйняття тексту. Для виключення строкатості сторінку рекомендується створювати з використанням не більше чотирьох основних кольорів.

На всіх сторінках одного сайту рекомендується використовувати однакові панелі навігації, розміщені в однакових місцях. Кожна Web-сторінка сайту повинна мати посилання на головну сторінку сайту.

На останній стадії проектування сайту доцільно перевірити працездатність всіх зроблених гіперпосилань. Завершуючи проектування сайту, бажано переглянути його за допомогою різних браузерів.

Коротко розглянемо питання оцінки допустимого обсягу Web-сторінки. У сучасному динамічному світі кожній людині потрібно встигнути багато зробити. Тому люди економлять кожну мить дарованої їм життя. Користувачі, які подорожують по мережі, "проскакують" повз тих Web-сторінок, які неприпустимо довго завантажуються на їх комп'ютери. З цієї причини при проектуванні сайту дизайнерові необхідно вміти заздалегідь оцінити приблизний час завантаження і допустимий обсяг створюваної ними сторінки.

Визначення в web-дизайні

Web-дизайнер - це фахівець, який займається проектуванням і розробкою web-вузла; створенням його структури, підбором кольорової палітри і розробкою користувальницького інтерфейсу.

Web-майстер - це фахівець, основним завданням якого є підтримка існуючого web-сайту. Підтримка полягає в інформаційному наповненні сторінок сайту і в віртуальній взаємодії з користувачами інтернету, які відвідують цей сайт.

Web-сайт - це сукупність web-сторінок, об'єднаних між собою єдиною композицією, темою інформаційного змісту, структурою, набором кольорів, і знаходяться на одному або декількох серверах.

Web-сторінка - це текстовий документ, складений з використанням мови HTML-розмітки. Web-сторінка є простим файлом з розширенням html або htm. Цей файл містить текст і зображення.

Браузер - це прикладна програма, яка дозволяє запитувати, отримувати з сервера і відображати web-сторінки.

Статичний web-вузол - це web-сайт, вміст якого фіксоване і не змінюється до тих пір, поки майстер не замінить дані на сайті.

Динамічний web-вузол - це web-сайт, в якому інформація зберігається в базі даних, а web-сторінки генеруються динамічно в відповідь на запит клієнта.

Сервер - це комп'ютер, підключений до мережі і призначений для управління ресурсами. Якщо він підключений до глобальної мережі інтернету, то на ньому можуть бути розміщені web-документи, які передаються у відповідь на запити клієнтів.

Електронна пошта - це система, призначена для обміну електронними повідомленнями з глобальної мережі інтернету.

Мережевий робот - це комп'ютерна програма, яка сканує різні web-сайти, витягує інформацію про ключові слова і зберігає її в базі даних пошукового сервера.

Пошуковий сервер - це комп'ютерна програма, яка організовує пошук за ключовими словами в глобальній мережі інтернету або на конкретному web-вузлі. У відповідь на запит користувача пошуковий сервер повертає список посилань на документи. Як приклад пошукового сервера можна назвати "google" і "yahoo".

Доменне ім'я - це назва web-сайту. Наприклад, www.ldubgd.edu.ua, www.site.com і інші.

Протокол інтернет (IP) - це протокол передачі інформаційних пакетів, який лежить в основі функціонування глобальної мережі інтернету.

Web-палітра - це набір з 216 кольорів. Web-орієнтована палітра гарантує, що кольорове зображення буде відображатися в незмінному вигляді більшістю браузерів, що працюють на різних платформах.

Протокол - це набір правил, що регламентують обмін інформацією між комп'ютерами в мережі. Як приклади мережевих протоколів можна привести IP, HTTP і FTP.

Гіпертекст - це інформаційна структура, яка забезпечує навігацію між web-документами за допомогою гіпертекстових посилань. Фрагменти тексту або зображення, при активації яких відображаються пов'язані з ними документи, називаються гіпертекстовими посиланнями.

Питання для самозакріплення:

- Що таке 'web-дизайн'?
- Дайте визначення 'web – сторінка'
- Назвіть допустимий об'єм web-сторінки
- Назвіть основні рекомендації при створенні web - сторінок

Рекомендована література:

- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.
- Cederholm, Dan. Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS. New Riders, 2007.
- Gasston, Peter. The book of CSS3: A Developer's Guide to the Future of Web Design. no starch press, 2011.

Лекція 2

Тема: Технології Web-дизайну. Частина 1

План заняття:

1. Логічна і фізична структура сайту
2. Верстка web-сторінок
 - а. Фіксований макет
 - б. «Гумовий» макет

Логічна і фізична структура сайту

Кожен ресурс Інтернету, від аматорської домашньої сторінки до великого інформаційного порталу, містить кілька тематичних рубрик, з'єднаних між собою гіперлінками. Як правило, посилання на всі розділи сайту з короткими анонсами їх вмісту наводиться на першій, так званій головній сторінці, якої присвоюється ім'я `index.html`.

ПРИМІТКА. Таке ім'я файлу рекомендується привласнювати всім стартовим документам сайту, оскільки в іншому випадку при зверненні до будь-якого розділу за допомогою скороченого URL без вказівки назви стартової сторінки (наприклад, `http://www.mysite.ua/photos/` замість `http://www.mysite.ua/photos/startpage.html`) браузер відобразить не саму web-сторінку, а перелік збережених в цій папці файлів.

Подібний набір тематичних категорій з розподіленими за відповідними розділами документами і заздалегідь спроектованими гіперлінками між усіма сторінками ресурсу і називається логічною структурою сайту. Фізична структура, навпаки, має на увазі алгоритм розміщення фізичних файлів по піддиректоріях папки, в якій опубліковано ваш сайт. Приклад порівняння логічної і фізичної структур одного і того ж ресурсу Інтернету показаний на рис. 2.1.

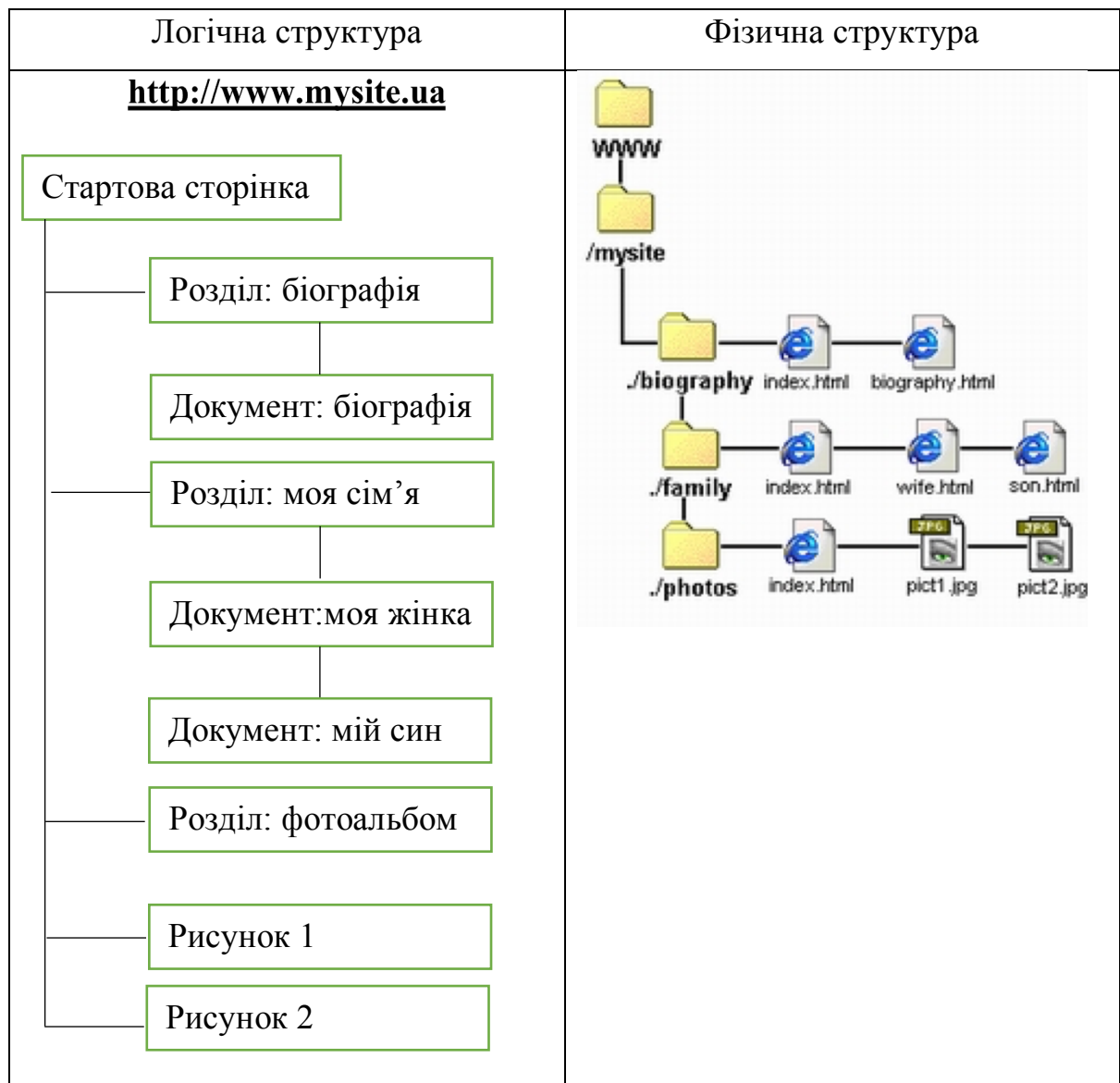


Рис. 2.1. Порівняння логічної і фізичної структури сайту

Очевидно, що логічна і фізична структури можуть не збігатися, оскільки в загальному випадку фізична структура ресурсу розробляється, виходячи зі зручності розміщення файлів. Однак більш-менш точне збереження порядку проходження логічних розділів у фізичній структурі

сайту дозволить вам уникнути плутанини при подальшому доповненні і оновленні матеріалів.

Порада, Рекомендується розміщувати всі графічні зображення, які є елементами проекту, в окремій папці з назвою "Images", розташованої в кореневій директорії сайту. Такий підхід дозволить оновлювати файли, котрі зберігаються в інших тематичних розділах без перенесення графіки, використовувати одні і ті ж графічні файли у всіх розділах сайту і при необхідності видаляти цілі директорії.

Для того щоб всі гіперпосилання на вашій домашній сторінці або web-сайті працювали коректно, всі документи відкривалися правильно і браузер не видавав помилок при зверненні до будь-яких розділах ресурсу, при створенні його фізичної структури слід дотримуватися кількох простих правил.

Порада. Призначайте імена директорій, імена і розширення документів HTML і графічних файлів з використанням символів тільки латинського алфавіту і тільки в рядковому реєстрі. Намагайтеся, щоб імена створених вами файлів і директорій не перевищували по довжині восьми символів.

Порада. При присвоєнні імен файлів документів HTML намагайтеся стежити за тим, щоб ці імена були "смысловими": згодом ви легко можете забути зміст і призначення будь-якої web-сторінки, якщо імена файлів будуть виглядати, наприклад, як 1.htm, 2. htm, 3.htm і т.д.

Для того щоб полегшити процес оновлення web-сторінок, додатки розділів або створення нових рубрик, заведіть засіб документування проекту: будь-яку електронну таблицю, створену, наприклад, в Microsoft Excel, або просто розграфлений зошит, куди записуйте відповідність елементів фізичної структури вашого проекту його логічної структури. До тих пір поки кількість складових вашого сайту (файлів) відносно мало. Приклад оформлення такого засобу документування показаний в табл. 2.1.

Таблиця 2.1. Приклад оформлення документування сайту

Ім'я файлу	Директорія	Опис	Дата створення / останньої зміни
index.html	/mysite	Стартова сторінка сайту http://www.mysite.ru	01.09.2018
index.html	/mysite/family	Стартова сторінка розділу <моя сім'я>	01.09.2018
wife.html	/mysite/family	Розповідь про мою жінку	02.09.2018
son.html	/mysite/family	Розповідь про мого сина	02.09.2018
sea.jpg	/mysite/photos	Фотографія з відпочинку на морі	03.09.2018

З усього сказаного стає очевидним, що фізична структура сайту прихована від відвідувачів вашого ресурсу: вони можуть спостерігати тільки логічну структуру, причому саме так, як вона представлена за допомогою елементів навігації. Звідси випливає цілком логічний висновок: будова системи навігації має якщо не повністю повторювати, то хоча б максимально відповідати розробленій вами логічній структурі сайту.

Головна сторінка

Один із критеріїв, керуючись яким можна розділити різні web-сайти на дві основні категорії, - це наявність головної сторінки (splash) або відсутність такої.

Головна сторінка являє собою html-документ, який не включає в себе будь-яку змістовну інформацію та елементи навігації. Файлу головної сторінки, як правило, присвоюється ім'я index.html. Головна сторінка містить, як правило, логотип компанії-власника даного ресурсу, лічильник відвідувань, вибір мови сайту, тощо. Приклад web-ресурсу, оснащеного такою сторінкою, показаний на рис. 2.2.



Рис. 2.2. Приклад сайту, оснащеного великою сторінкою

Верстка web-сторінок.

Веб-сторінка запускається на комп'ютері під керуванням клієнтської програми - браузером. Зрозуміло, що операційна система, її налаштування і власне сам браузер відрізняється у різних користувачів. З чого напрошується банальний висновок, що один і той же сайту по-різному відображається у кожного користувача.

Напрошується питання, чи можна зробити так, щоб веб-сторінка відображалася однаково? Тому завдання верстки веб-сторінок формулюється так: сформуванати документ, який би коректно відображався з невеликими відмінностями на основних платформах і в браузерах.

Щоб реалізувати зазначене завдання треба розуміти, як взагалі відбувається верстка веб-сторінок і порівнювати свої ідеї з їх виконанням.

Особливості верстки документів сайту.

Ширина документу

Спочатку розробникові сайту ширина вікна браузера користувача невідома, оскільки вона може змінюватися в довільних межах. Ширина залежить від розширення монітора, довжини його діагоналі, розміру вікна і ще деяких варіюваних даних. Іншими словами передбачити її заздалегідь простими засобами не представляється можливим. З урахуванням цієї особливості утвердилося два способи верстки: фіксований і «гумовий».

Фіксований макет.

В даному випадку діємо від зворотнього і встановлюємо загальну ширину макету жорстко заданій і рівній певній величині. Якщо взяти деяку узагальнену статистику відвідувачів сайтів і подивитися, яке розширення монітора вони переважно використовують, то дізнаємося, що це 1024 x 768 пікселів. Виходить, що ширина монітора користувачів в основному 1024 пікселя. Візьмемо за орієнтир 1014 пікселів, тоді загальна ширина макету за вирахуванням вертикальної смуги прокрутки і рамки браузера опиниться 980-990 пікселів. На цей розмір орієнтуємося і встановлюємо ширину макету, наприклад 980 пікселів.

Перевага такої схеми наступна. Якщо загальна ширина макету точно відома, то ми можемо легко підганяти під неї дизайн і робити зображення вже відомої ширини. В цілому подібна верстка схожа до верстки друкарського буклета, і в тому і в іншому випадку ширина носія інформації суворо задана, за рахунок чого верстка хоч частково, але спрощується.

Недолік, який інкримінують цьому виду верстки фактично один - недостатньо ефективне використання вільної площі. Дійсно, для монітора з великою діагоналлю або високим розширенням екрану документ виглядатиме по-іншому, ніж на передбачуваних 1024 пікселях. На рис. 2.3 показано, як виглядає макет в такому випадку.



Рис. 2.3. Макет фіксованої ширини з розміщенням по лівому краю

Справа з'являється широка порожня смуга, розмір якої залежить від розширення екрану користувача і діагоналі його монітора. Щоб хоч якось зменшити порожній простір, макет зазвичай поміщають по центру вікна браузера (рис. 2.4).



Рис. 2.4. Макет фіксованої ширини з розміщенням по центру
«Гумовий» макет.

Цей вид макета ґрунтується на тому, що в якості однієї з одиниць вимірювання виступають відсотки. Спільна робоча ширина вікна браузера - 100%, і колонки макету в сумі не повинні її перевищувати, тому для зручності, як правило, скрізь застосовують процентний запис. При зміні розмірів вікна відбувається переформатування даних сторінки, щоб вони вписалися в нову ширину (рис. 2.5).

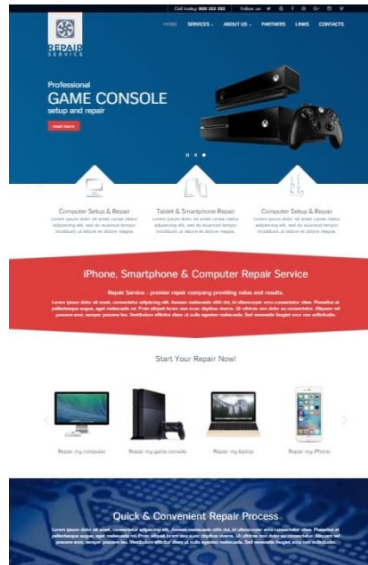


Рис. 2.5. Веб-сторінка, котра займає всю відведену ширину

На сьогоднішній день цей вид верстки є основним. Але слід відразу зазначити деякі його особливості та недоліки:

- хоч веб-сторінка і підлаштовується під ширину вікна браузера, при досягненні певної величини читати текст стає незручно - рядки дуже довгі і очі втомлюються по ним бігати. Втім, браузер можна згорнути в вікно, підбравши його комфортний розмір;

- верстати «гумовий» макет складніше, ніж аналогічний, але фіксованої ширини. Це пов'язано з тим, що доводиться враховувати безліч додаткових чинників і знати деякі прийоми верстки. До того ж, популярні браузери неоднозначно трактують деякі параметри і в них «гумовий» макет може відобразитися по-різному;

- будь-який макет має деяку мінімальну ширину, при досягненні якої веб-сторінка «розсипається» або з'являється горизонтальна смуга прокрутки. Наприклад, якщо в документ вставлений малюнок шириною 600 пікселів, то при зменшенні вікна до цієї величини браузер почне відображати смугу прокрутки.

Питання для самозакріплення:

- Чим відрізняється логічна і фізична структура сайту.
- Дайте визначення «фіксованому» макету.
- Дайте визначення «резиновому» макету.

Рекомендована література:

- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.
- Cederholm, Dan. Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS. New Riders, 2007.
- Gasston, Peter. The book of CSS3: A Developer's Guide to the Future of Web Design. no starch press, 2011.

Лекція 3

Тема: Технології Web-дизайну. Частина 2

План заняття:

1. Прямокутні об'єкти веб-сторінки
2. Модульна сітка

Прямокутні об'єкти веб-сторінки

Всі об'єкти на веб-сторінці мають прямокутну форму. Цей простий постулат не завжди узгоджується з тим, що ми бачимо в дійсності, тому слід зробити пояснення. На відміну від традиційної верстки (мова йде про поліграфічні матеріали), де в документ можна вставляти будь-які об'єкти, в тому числі і векторні фігури довільної форми, верстка веб-сторінок має ряд обмежень. До обмежень відноситься і те, що додаються об'єкти прямокутні. Причому це відноситься до їхніх форм, але не до вмісту, завдяки чому необхідний нам дизайн можна конструювати за допомогою набору зображень. На рис. 3.1 показана картинка доволі неправильної форми. Але через те, що фон у цього малюнка збігається з фоном веб-сторінки, вихідна прямокутність зображення не видно.



Рис. 3.1 - Зображення може мати довільну форму

Активне використання зображень

Зображення не тільки застосовуються для ілюстрації тексту, а й виконують на сайті багато різних ролей, наприклад, використовуються для створення привабливого дизайну, служать розпіркою між елементами таблиці, створюють градієнтні заливки, фонові зображення і т.д.

Розрізання зображення на фрагменти

Один малюнок може займати дуже велику вихідну площу. Розрізавши його на прямокутники, отримаємо чудовий конструктор, в якому одні фрагменти зображення допускається замінювати текстом, інші анімацією, а треті модифікувати «на льоту». Таким чином, маємо в своєму розпорядженні засіб для обходу прямокутної природи зображень, адже в «склеєному» малюнку може бракувати деяких фрагментів.

Фоновий малюнок зручний тим, що він може заповнювати всю відведену ширину або висоту під блоком. Це дозволяє створювати лінії або інші декоративні елементи, які прив'язуються до ширини або висоти тексту і не залежать від розмірів вікна. До того ж поверх фону можна накладати текст, що також розширює можливості по дизайну веб-сторінок.

Зображення замість тексту

Якщо засоби верстки мають певні обмеження, то чому б не створити текст в графічному редакторі і не вставити його в якості картинки або Flash? Це гарантує, що текст збереже свій вигляд незважаючи ні на які зовнішні умови. Однак тут є і зворотний бік - малюнки займають більший об'єм, ніж рядовий текст, їх складніше правити, вони не індексуються пошуковими машинами, їх показ користувачі можуть відключити. Загалом, недоліків маса, тому зображення на сайті хоча і застосовують замість тексту, але досить обмежено. Наприклад, для заголовка сайту.

Модульна сітка

Модульна сітка являє собою набір невидимих направляючих, уздовж яких розташовуються елементи веб-сторінки. Це полегшує розміщення даних в документі, забезпечує візуальний зв'язок між окремими блоками і зберігає спадкоємність дизайну при переході від однієї сторінки до іншої.

Веб-сторінка фактично розглядається як набір прямокутних блоків, які викладаються в певному порядку. При цьому, як правило, дані розташовуються по колонках, тому при верстці застосовують термін одно-, дво-, трьохколонний макет і т.д. Для прикладу розглянемо головну сторінку сайту 'computer service' (рис. 3.2)



Рис. 3.2 - Титульна сторінка сайту 'computer service'

Кожен блок цієї сторінки чітко відділений від інших за допомогою порожнього простору або рамки. Напрошується тільки питання, чи дійсно ми маємо справу з трьома колонками?

Слід розуміти, що колонки можуть об'єднуватися, а також містити не тільки суцільний текст, а й графічні вставки. Якщо уявити основні блоки сторінки у вигляді однотонних прямокутників, то отримаємо наочну модульну сітку, по якій відразу стає зрозуміло, як зверстаний документ (рис. 3.3).

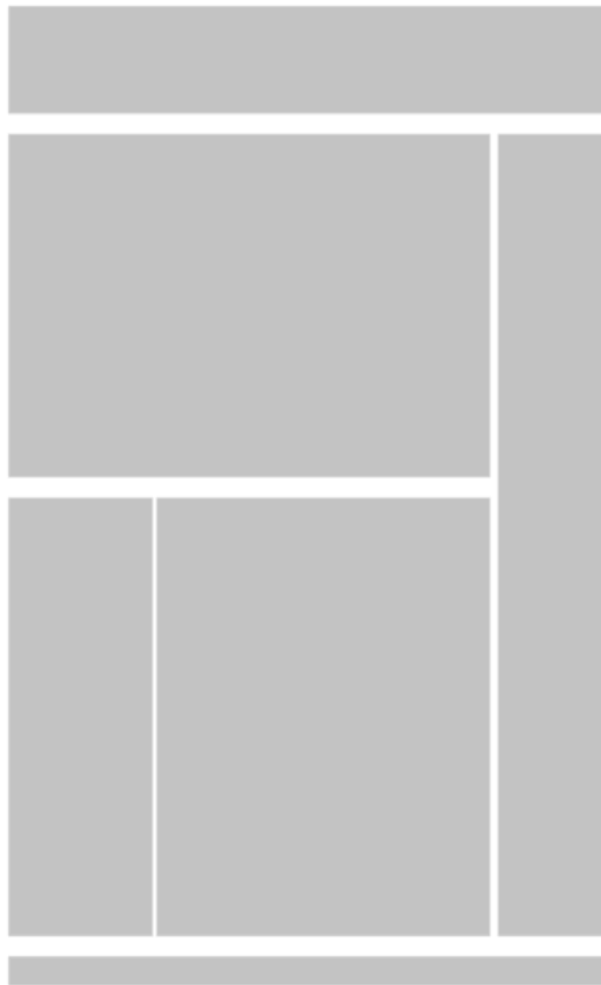


Рис. 3.3. – Приклад модульної сітки сайту

За даним рисунком видно, що верхній блок з назвою сайту, формою пошуку і кнопками навігації займає всю ширину сторінки. Далі йдуть три колонки, причому перші дві колонки попередньо об'єднані в одну для зручного представлення необхідної інформації. Завершує макет блок з контактною і правовою інформацією.

Перейдемо до принципів побудови модульної сітки. Спочатку макет веб-сторінки розробляють на аркуші паперу. Це дозволяє, не витрачаючи даремно часу, швидко зробити серію нарисів і вже з них вибрати відповідний ескіз. Скільки варіантів можна створити за десять хвилин в графічному редакторі і скільки за цей же час за допомогою олівця і паперу? При цьому не має значення ступінь володіння програмою, все одно на папері вийде швидше. Зазвичай замість тексту і малюнків застосовують схематичні значки. Наприклад, текст позначається декількома горизонтальними лініями (рис. 3.4), а малюнки зображаються затемненими блоками або покарбованими прямокутниками (рис. 3.5).



Рис. 3.4 – Позначення тексту в макетах



Рис. 3.5. Позначення зображень в макетах

Одноколонна сітка

Текст в одну колонку найчастіше зустрічається в академічному дизайні, при фіксованому макеті і публікації великої тексту.

На рис. 3.6 показана типова схема одноколонної модульної сітки. Як правило, спостерігається чотири основні блоки - заголовок сторінки, набір посилань на інші сторінки сайту (навігація, меню), власне сам текст і в самому низу контактна інформація («підвал» сторінки). Якщо висота сторінки досить велика, то блок навігації дублюють внизу або роблять посилання «Вгору», яка переміщує до початку документа.

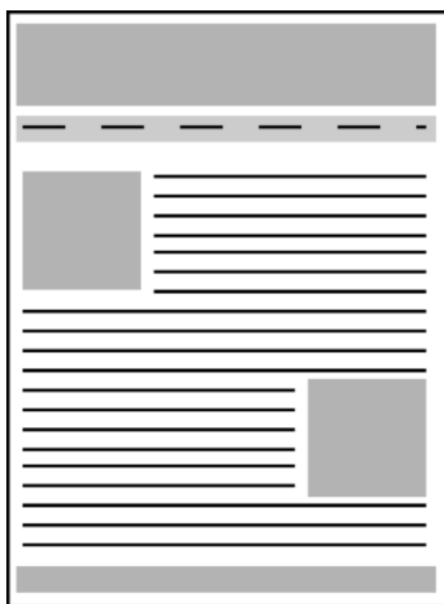


Рис. 3.6 - Одноколонна модульна сітка

Текст зазвичай обтікає ілюстрації по контуру. При активному застосуванні зображень на сайті зручніше скористатися фіксованим макетом, ширина якого точно відома. Тоді малюнки можна готувати заданого розміру, які точно впишуться в макет сторінки фіксованої ширини і практично повністю складається з набору малюнків.

[Двоколонна сітка](#)

Це один з найпоширеніших варіантів при використанні на сайтах. При такій модульній сітці використовується дві колонки - одна відводиться під основний текст, а друга використовується для навігації і іншої корисної інформації (рис. 3.7).



Рис. 3.7 – Двохколонна модульна сітка

Принципового значення не має, зліва чи справа розташовується колонка з навігацією, зустрічається і той і інший варіант.

Двоколонні сітки зручні при створенні найрізноманітніших сайтів і не вимагають особливих знань по верстці веб-сторінок. Єдиний недолік, який їм закидають, що подібні сайти виглядають досить одноманітно. Але з іншого боку користувачам зручніше працювати з сайтом звичного вигляду, без зайвих «наворотів».

Трьохколонна сітка

Такі сітки часто застосовуються на головних сторінках сайтів, де одночасно потрібно показати користувачеві безліч можливостей, які він виявить на даному сайті. Також трьохколонна сітка використовується і на внутрішніх сторінках, якщо для розміщення різної інформації двох колонок вже не вистачає (рис. 3.8).



Рис. 3.8 – Трьохколонна модульна сітка

Одна з колонок віддається під навігацію, друга, найширша - під основний текст, а в третю колонку додають рекламу, посилання, текст і т.д.

Трьохколонна сітка забезпечує більше простору для дизайну, адже в деяких місцях можна об'єднувати колонки, розбивати матеріал на окремі фрагменти і візуально відокремлювати один блок від іншого. Макет при цьому може вийти досить складним, але результат зазвичай того варто.

До недоліків трьохколонної сітки відноситься відносно висока складність верстки макету. Щоб отримати потрібний результат доводиться іноді затратити багато часу на створення стильового файлу і налагодження документа в різних браузерах.

Звичайно, існують і інші види модульних сіток, наприклад, багатоколонні. Однак вони практикуються досить рідко через обмежену ширину вікна браузера. При такому розкладі текст доводиться робити дрібним або застосовувати інші способи, щоб не виникло горизонтальної смуги прокрутки. Від цього може постраждати зручність сприйняття інформації користувачем.

Модульна сітка не є єдиним вірним варіантом при верстці веб-сторінок. Можна скористатися перспективою, хаотичністю або іншою структурою, головне, щоб вона була естетичною і привабливою. Відмова від правил верстки характерний для дизайн-студій, завдання яких полягає в тому, щоб вразити своїм сайтом увагу відвідувачів і привернути до себе їх увагу. Але навіть вони роблять для замовника сайти на основі формальних сіток, оскільки так простіше додавати інформацію і вносити різноманітність в дизайн макету.

Модульні сітки дозволяють спростити верстку сайту, оскільки всі матеріали розбиваються на окремі блоки, які вирівнюються по невидимим лініях. Такі блоки хоча і взаємопов'язані між собою, але зазвичай дозволяють незалежне редагування даних, що спрощує оформлення елементів.

Питання для самозакріплення:

- Дайте визначення «резиновому» макету.
- Дайте визначення «фіксованому» макету.
- Чим відрізняється фізична та логічна структура сайту.

Рекомендована література:

- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

- Cederholm, Dan. Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS. New Riders, 2007.
- Gasston, Peter. The book of CSS3: A Developer's Guide to the Future of Web Design. no starch press, 2011.

Лекція 4

Тема: Принципи компоювання і алгоритм створення web- сайту

План заняття:

3. Принципи компоювання web-сайту
4. Статичне і динамічне компоювання сторінки
5. Елементи web – сторінки

Принципи компоювання web - сайту

Сучасні відеокарти підтримують кілька довільно встановлюваних відеорежимів, що характеризуються екранним дозволом і кількістю кольорів, використовуваних при відображенні інформації на екрані комп'ютера. За допомогою вбудованих функцій операційної системи користувач може встановити кілька стандартних значень екранного дозволу, наприклад 640x480, 800x600, 1024x768, 1152x864, 1280x1024, 1600x1200 точок, тощо. Зрозуміло, при відкритті в одному і тому ж браузері будь-якої web-сторінки вона може відобразитися однаковим чином в залежності від використовуваного відвідувачем сайту екранного дозволу.

Для побудови складних таблиць можна застосовувати вкладення однієї таблиці в іншу: наприклад, первинна таблиця шириною 640 точок задає горизонтальне розбиття сторінки, в кожену з її осередків вкладається ще одна, рівна їй по ширині невидима таблиця, що забезпечує вертикальне розбиття. Такий підхід дозволяє дробити різні ділянки web-сторінки незалежно один від одного, створюючи, наприклад, в одному її полі чотири вертикальні колонки різної ширини, а в іншому - дві, оперуючи при цьому різними параметрами вкладених таблиць в межах одного і того ж документа - розміром простору між колонками, товщиною кордонів і т.д.

Такий варіант компоювання сайту можна назвати статичним, оскільки ширина таблиці не змінюється в залежності від екранного розширення. Зрозуміло, при зміні параметрів екрану не відбувається ні найменшого зміщення елементів дизайну сторінки.

Інший підхід - коли ширину невидимої таблиці, яка містить фрагменти web-сторінки, задають у відсотках від поточної ширини екрану. При збільшенні екранного розширення таблиця "розтягується" по горизонталі, і всі розміщені в її осередках елементи, позиційні або по центру, або по краях стовпців, зміщуються відповідно до встановленого алгоритму. В силу того, що параметри таблиці змінюються в залежності від налаштувань екрану, такий принцип компоювання html-документа можна

назвати динамічним. І той і інший підхід має як переваги, так і недоліки, які перераховані нижче.

Статична компоновка сторінки

Переваги. Простота алгоритму верстки документа. Оптимізувавши сторінку для відображення з роздільною здатністю 640x480 пікселів, ви можете бути впевнені, що при зміні користувачем екранних налаштувань елементи дизайну не "попливуть". Крім того, даний варіант компоновки сайту в більшості випадків (при дотриманні ряду додаткових умов) дозволяє домогтися ідентичності відображення html-документа в браузерах.

Недоліки. При відображенні документа на комп'ютері з високим екранним розширенням по краях екрану або з правого його боку залишається помітне порожнє поле.

Динамічне компоновка сторінки

Переваги. Документ розтягується по всій ширині екрану, не залишається невикористаних порожніх полів.

Недоліки. Складність верстки та налагодження сторінки, досить часто виявляється неадекватність відображення таких документів в браузерах Microsoft Internet Explorer і Netscape Navigator.

Приклади реалізації сайтів зі статичним та динамічним компоновкою сторінок наведені на рис. 3.1 і 3.2. Дані зображення зроблені з фіксованим екраном роздільною здатністю 800x600 пікселів.

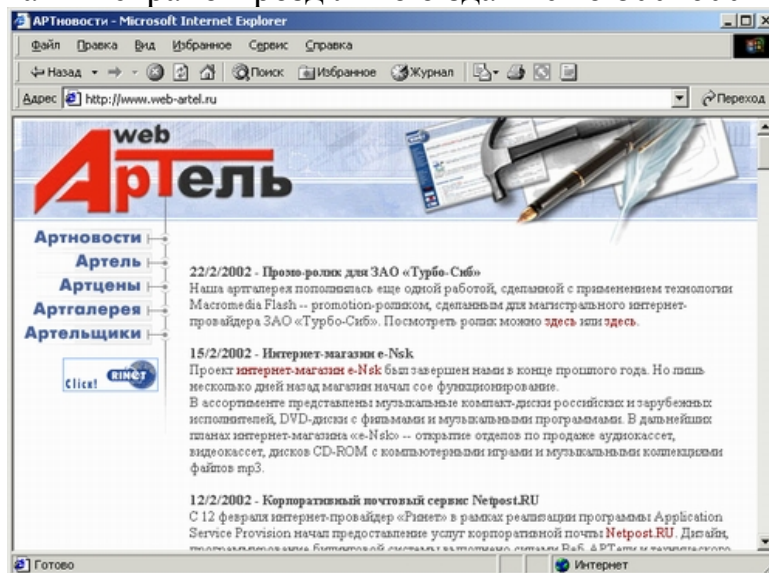


Рис. 3.1. Пример сайта со статичной компоновкой страниц



Рис. 3.2. Приклад сайту з динамічним компонуванням сторінки

Елементи web-сторінки

Будь-яка web-сторінка містить певний набір стандартних елементів, які є обов'язковими компонентами кожного ресурсу Інтернету. Безумовно, асортимент і кількість подібних об'єктів можуть варіюватися в залежності від тематичної спрямованості сайту, обсягу опублікованих на ньому матеріалів, а також від цілей і завдань, які ставить перед собою творець даного ресурсу. Компонування таких елементів, проектування їх взаємного розташування і становить одну з головних завдань web-майстрів.

Першим елементом web-сторінки, який нам належить розглянути, є її заголовок. Він може бути виконаний як в текстовому, так і в графічному вигляді, але і в тому і в іншому випадку він повинен розташовуватися у верхній частині документа. Іноді з заголовком поєднують меню вибору кодування кирилиці і кнопки для переходу з україномовної на англійську мовну версію сайту, якщо даний web-ресурс представлений на двох мовах. Безпосередньо під заголовком документа, як правило, розташовується простір, відведений для розміщення рекламного банера. Включення банера саме в верхню частину web-сторінки в більшості випадків є обов'язковою умовою реєстрації сайту в службах банерного обміну - системах, які рекламують створений вами ресурс в обмін на показ на сторінках вашого сайту реклами інших учасників банерообмінної мережі. Стандартний розмір банерів, що публікуються під заголовком документа, становить зазвичай 468x60. Якщо ви використовуєте статичний принцип компонування сторінки, ширина заголовка вашого документа буде складати приблизно 640 пікселів: це значення обумовлене, перш за все, необхідністю забезпечити коректне відображення документа на моніторах з екранною роздільною здатністю 640x480 пікселів і уникнути появи горизонтальної смуги прокрутки, що утрудняє його перегляд. Очевидно, що

ширина банера в цьому випадку буде значно менше ширини заголовка, завдяки чому в тій частині сторінки, де ви плануєте відвести місце під рекламу, утворюється незайнятий простір, який можна заповнити логотипом компанії-власника даного сайту або посиланням на сервер, який здійснює web- хостинг. Зрозуміло, логотип необхідний далеко не завжди: як правило, він включається до складу web-сторінки лише в разі, якщо сайт має комерційну спрямованість.

Основну частину документа займає так зване текстове поле - ділянка, де і розміщується смислове наповнення сторінки: змістовний інформаційний текст і ілюстрації. Перераховані елементи ще називають "контент" (від англ, content - зміст). Розташування текстового поля залежить в першу чергу від того, яким чином web-дизайнер розмістить інші елементи документа.

Наступною обов'язковою складовою частиною web-сторінки є елементи навігації - гіперпосилання, що зв'язують даний документ з іншими розділами сайту. Елементи навігації можуть бути виконані у вигляді текстових рядків, графічних об'єктів, тобто кнопок, або активних компонентів, наприклад Java-апплетів. Останні являють собою ті ж кнопки, які, на відміну від своїх "традиційних" сестер, вміють реагувати на рухи миші, виконуючи при наведенні на них курсору будь-які нескладні дії (включення підсвічування, створення ефекту "натискання", зміна форми і т. д.). Найбільш усталеним підходом є розміщення елементів навігації в лівій межі сторінки.

У нижній частині документа прийнято публікувати інформацію про розробників сайту та адресу електронної пошти, за яким відвідувачі ресурсу можуть направити власникам сторінки свої відгуки, пропозиції і побажання. Якщо web-сторінка є стартовим документом, в нижній її частині також розміщують лічильник відвідувань. Не рекомендується також розміщувати на одній сторінці кілька різних лічильників.

Приклад компоновання сайту, що містить повний набір описаних вище складових, показаний на рис. 3.3. У ньому вибрано позиціонування елементів навігації по лівій межі документа.

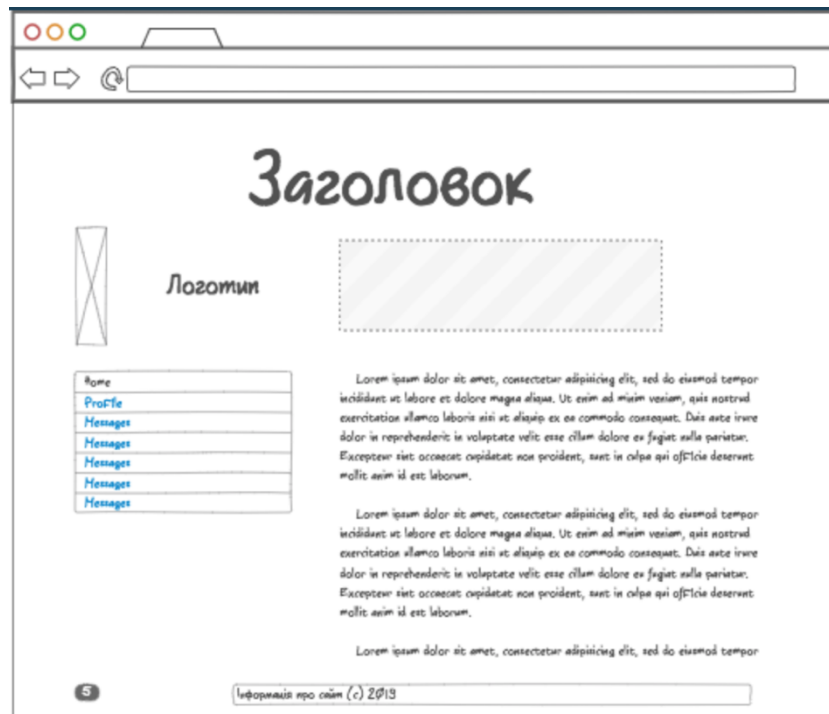


Рис. 3.3. Приклад компоновання web-сторінки з лівим позиціонуванням елементів навігації

На практиці часто зустрічаються web-сайти, в дизайні яких елементи навігації позиціоновані по правій стороні екрану. В цьому випадку текстове поле зміщується ліворуч, інші компоненти документа розташовуються, виходячи з принципу максимальної естетичності їх поєднання. Приклад такого виконання сайту показаний на рис. 3.4.

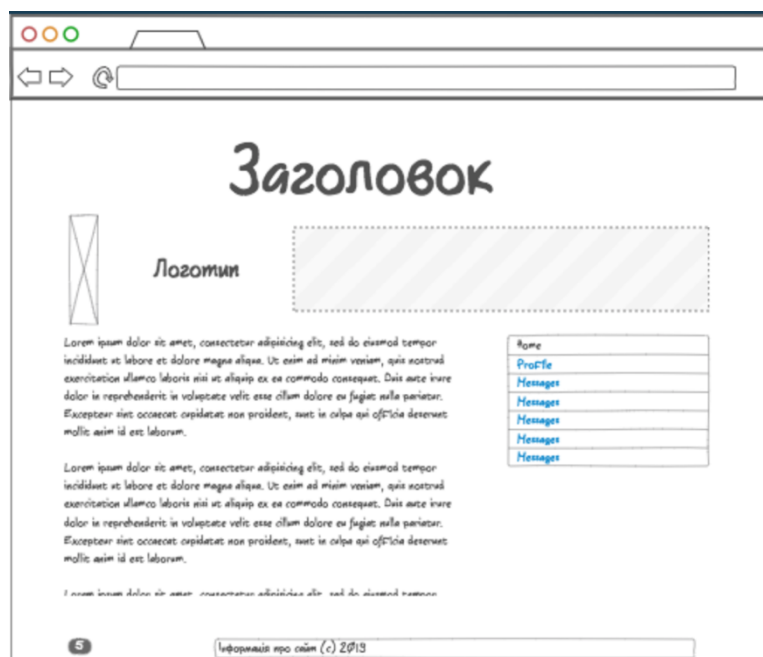


Рис. 3.4. Приклад компоновання web-сторінки з правим позиціонуванням елементів навігації

При такому підході рекомендується витримувати графічне оформлення заголовка, логотипу та поля для розміщення реклами в єдиному колірному і художньому стилі - тоді несиметричність становища даних об'єктів буде не настільки очевидна і не стане "різати око" прихильникам суворої, "табличної" естетики дизайну.

Елементи навігації можна розмістити не тільки поблизу правої і лівої межі сторінки, а й у верхній частині документа. Приклад оформлення сторінки з верхнім розташуванням елементів навігації показаний на рис. 3.5.

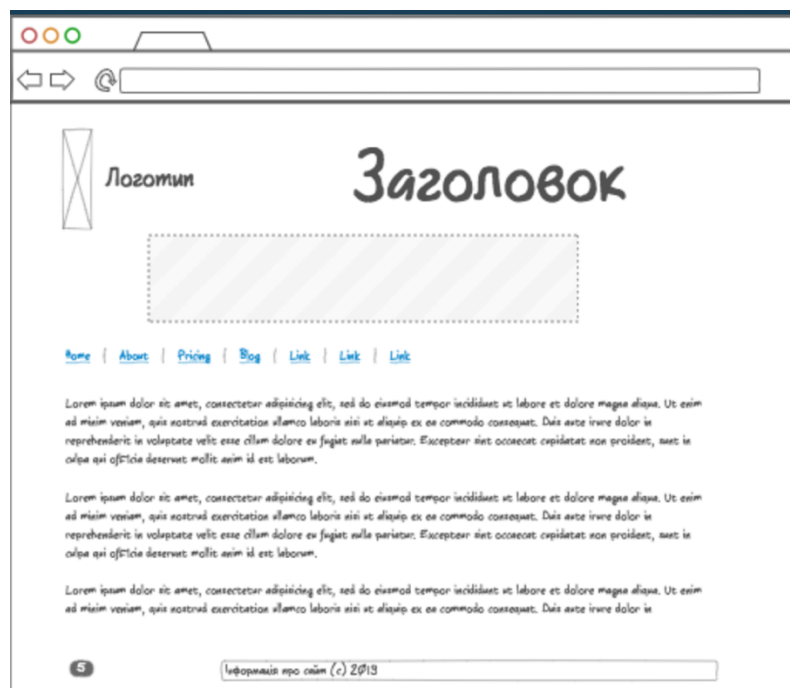


Рис. 3.5. Приклад компоновання web-сторінки з верхнім позиціонуванням елементів навігації

Очевидно, що варіантів "змішаного компоновання" web-сторінки може бути безліч: конкретні рішення залежать від кількості складових ресурсу, обсягу підготовленого для розміщення на сайті тексту і, нарешті, від фантазії самого дизайнера.

Питання для самозакріплення:

- 1) Що таке динамічна компоновка сайту?
- 2) Що таке статична компоновка сайту?

3) Назвіть основні елементи web- сторінки.

4) Назвіть класифікацію web-сайтів за ознаками їх перегляду.

Рекомендована література:

- Cederholm, Dan. Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS. New Riders, 2007.
- Gasston, Peter. The book of CSS3: A Developer's Guide to the Future of Web Design. no starch press, 2011.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 5

Тема: Графічні зображення. Особливості веб-графіки. Частина 1

План заняття:

6. *Графіка для web*
7. *JPEG*
8. *GIF*
9. *PNG*

Графіка для web

Оформлення Web-сторінок – мається на увазі необхідність розміщення максимально можливого обсягу інформації на порівняно невеликому просторі.

Готуючи графіку для Web-сторінки, необхідно пам'ятати про невеликі розміри області, з якою ви повинні працювати. Зображення не повинно перевищувати 700 пікселів по ширині і 400 пікселів по висоті. Але файл навіть такого розміру може виявитися занадто великим для передачі по каналах зв'язку мережі. Слід також пам'ятати і про те, що частина користувачів можуть бачити на своїх моніторах тільки 256 кольорів. Тому

24-бітна графіка, здатна забезпечити 16,7 мільйона відтінків кольору, не зможе бути оцінена деякою частиною глядачів. При цьому користувачі не тільки не зможуть побачити на своїх моніторах всієї закладеної вами колірної гами, але, крім того, зображення може виявитися гіршої якості, ніж в разі використання 8-бітної графіки, що відображає 256 кольорів. До того ж файл виявиться в три рази більше за розміром, так як, його розмір пропорційний глибині кольору, і, відповідно, буде вимагати більшого часу для завантаження на екран.

Використання коректної колірної гами важливо при розробці будь-якого графічного проекту. Для графіки Web-сторінок необхідно застосовувати кольори моделі RGB. Так як для Web-сторінок пристроєм виведення графіки є екран монітора, то тут, як правило, виконується умова WYSIWIG - What You See Is What You Get (Що бачиш, те й отримуєш). Говорячи «як правило», ми маємо на увазі невеликі індивідуальні відмінності моніторів користувачів.

На питання «Якою має бути розмір графічного файлу?», Призначеного для Web, важко відповісти однозначно. Давайте спробуємо уявити собі, що користувач, який звертається до вашого сайту, має модем з пропускною спроможністю 144,4 Кбіт / сек, хоча подібні модеми і вважаються застарілими. При використанні такого модему передача блоку даних об'ємом 18 кілобайт займає приблизно 1 секунду, так що при обсязі графіки в 180 кілобайт для її передачі по мережі потрібно приблизно 10 секунд. Після таких розрахунків виникає наступне питання: як довго відвідувач, який звернувся до вашого сайту, готовий чекати появи на екрані завантажуваного зображення? Якщо ви зможете відповісти на це питання, то відразу ж визначте і максимальний розмір файлу з графічним зображенням. Вважається, що файл повинен мати такий розмір, при якому час його завантаження по мережі не перевищує десяти секунд.

Розмір графічного файлу, зокрема, отриманого в результаті сканування зображення, можна визначити за такою формулою:

$$F = (A/25,4 \times B/25,4) \times D^2 \times C/8,$$

де

F - розмір файлу без стиснення в байтах;

A - ширина вихідного зображення в мм;

B - довжина вихідного зображення в мм;

D - графічне дозвіл зображення в dpi;

C - глибина кольору в бітах.

Таким чином, якщо ви хочете відсканувати для своєї Web-сторінки фотографію розміром 10x15 см з роздільною здатністю 150 dpi і глибиною кольору 24 біта, то результуючий файл буде мати розмір 1569378 байт або 1,49 мегабайта.

Будь-яка комп'ютерна інформація може зберігатися тільки в певному форматі. Кожен вид інформації має власні формати. Для тексту використовуються одні формати, для електронних таблиць - інші, для графіки треті. Формат графічної інформації зазвичай визначається програмою, в якій вона створена. Для Web-графіки стандартними вважаються формати JPEG, GIF і PNG.

JPEG

Формат JPEG (Joint Photographic Experts Group - - найпопулярніший формат для зберігання фотографічних зображень. JPEG є стандартом в Інтернеті і призначений для збереження растрових або точкових зображень зі стисненням, що зменшує розмір файлу від десятих часток відсотка до 100 разів, за рахунок відкидання надлишкової інформації, що не впливає на відображення документа. Але практично діапазон стиснення значно вузький: від 5 до 15 разів. Хоча алгоритм стиснення зображення цього формату і погіршує якість, він може бути легко налаштований на

мінімальні, практично непомітні для людського ока, втрати. Розпакування JPEG-файлу відбувається автоматично, під час його відкриття. Між якістю зображення і ступенем ущільнення існує зворотна залежність: чим вищий ступінь стиснення для результуючого зображення ви задасте, тим гірше буде його якість.

Існує різновид формату JPEG, іменована «прогресивний JPEG» (p-JPEG). Прогресивний JPEG відрізняється від звичайного тим, що при виведенні на екран зображення з'являється майже відразу, але поганої якості, а в міру завантаження якість поліпшується. У форматі JPEG можна зберігати тільки повнокольорні зображення з глибиною кольору 24 пікселя. Існує також версія JPEG для 32-бітних зображень, але вона поки мало поширена. Зазвичай формат JPEG рекомендують використовувати для фотографій, на яких зниження якості, супутнє стиску за цією технологією, найменш помітно.

GIF

Формат GIF (Graphics Interchange Format - Формат для обміну графічною інформацією був розроблений фахівцями комп'ютерної мережі CompuServe в 1987 році з метою прискорення пересилки файлів по телефонних лініях. Необхідно було створити такий формат, який би містив досить добре упаковані дані. Алгоритм розпакування повинен був бути при цьому вбудований в усі програми, які будуть працювати з цим форматом: утиліти перегляду, графічні редактори і т.п. Саме такий підхід і реалізований у форматі GIF.

Хоча з тих часів пропускна здатність модемів збільшилася в багато разів, вимоги до компактності файлів, переданих по мережних каналах, аж ніяк не відійшли на другий план. Web-сторінка, яка з усім своїм текстом і графікою займає понад 40-50 кілобайт, вже вважається «великою» - і далеко не кожен відвідувач дочекається, поки вона повністю завантажиться в його браузер. Професіоналізм Web-дизайнера як раз і полягає в тому, щоб

відшукати належний баланс між художністю, інформативністю сторінки і її об'ємом. Ось чому формат GIF, головною перевагою якого є малий розмір файлів, до сих пір зберігає своє значення основного графічного формату World Wide Web. З поширених графічних форматів GIF однозначно поступається за ступенем стиснення тільки формату JPEG.

Звичайно, GIF - не єдиний формат, що дозволяє стискати графічні дані; деякі здатності до стиснення є у форматів BMP, PCX, TIFF і багатьох інших. Однак GIF - один з небагатьох форматів, що використовує алгоритм стиснення, майже не поступається за ефективністю програмам-архіваторам. Іншими словами, GIF-файл не потрібно архівувати, так як це рідко дає суттєвий вииграш в обсязі.

Як і у програм-архіваторів, ступінь стиснення графічної інформації в форматі GIF сильно залежить від рівня її повторюваності і передбачуваності, а іноді ще й від орієнтації картинки. Оскільки GIF сканує зображення по рядках, то, наприклад, плавний перехід кольорів - градієнт, спрямований зверху вниз, стиснеться куди краще, ніж тих же розмірів градієнт, орієнтований зліва направо, а цей останній - краще, ніж градієнт по діагоналі.

На відміну від формату JPEG, який дозволяє досягти серйозного виграшу в розмірі при мінімально помітних для ока втратах в якості, формат GIF користується звичайним стисненням без втрат. З іншого боку, перевагою JPEG в порівнянні з GIF є можливість зберігати повнокольорові зображення з 16 мільйонами кольорів, тоді як GIF обмежений лише 256-кольоровий палітрою.

У 1989 році з'явилася нова, вдосконалена версія формату GIF, в якій використовується «поступовий» показ зображення під час його завантаження з мережі. Ця технологія полягає в тому, що при показі спеціально записаних GIF-файлів рядки зображення виводяться на екран не підряд, а в певному порядку: спочатку кожна 8-ма, потім 4-я і т.д., тобто

повністю зображення показується в чотири підходи. Така система показу дозволяє зрозуміти, що зображено на картинці ще до того, як вона повністю завантажиться, що дуже важливо на повільних лініях зв'язку. Щось подібне використовується в телебаченні для передачі одного повного кадру за два проходи променя.

PNG

Формат PNG (Portable Network Graphics). Цей формат, оснований на варіації алгоритму стиснення без втрат якості, на відміну від GIF стискає растрові зображення і по горизонталі, і по вертикалі, що забезпечує більш високий ступінь стиснення; крім того, він підтримує кольорові графічні зображення з глибиною кольору до 48 біт включно.

PNG дозволяє зберігати повну інформацію про ступінь прозорості в кожній точці зображення у вигляді так званого альфа-каналу. Кожен піксель PNG-файлу, незалежно від його кольору та розташування, може мати будь-яку градацію прозорості: від нульової - повна непрозорість, до абсолютної невидимості.

Щоб згладити відмінності в яскравості дисплеїв, що існують між різними типами комп'ютерів і, відповідно, між створюваними на них зображеннями, PNG дозволяє зберігати в файлі вихідний коефіцієнт яскравості того дисплея, на якому зображення було створено. Це дає можливість програмам перегляду перед виведенням на екран регулювати яскравість зображення, пристосовуючи її до особливостей даної відеосистеми - так звана гамма-корекція.

PNG-формат не дає можливості створювати анімаційні ролики. Тому для зберігання анімації або декількох зображень в одному файлі використовується варіація формату PNG - формат MNG (Multiple Network Graphics). В даний час розробники формату PNG працюють над новим форматом PNP (пінап) (Portable Network Photography), який призначений

для зберігання фотографічних зображень і володіє потужним алгоритмом стиснення з втратою якості.

Питання для самозакріплення:

- 1) Які види графічних форматів вам відомі?
- 2) Який об'єм графічного файлу допустимий на веб-сторінці?
- 3) Чим відрізняється GIF від JPG?

Рекомендована література:

- Wahyudin, A. (2019). IMPLEMENTASI RESPONSIVE WEB DESIGN PADA SISTEM INFORMASI GEOGRAFIS OBJEK CENTRAL BIMA (NTB) (Doctoral dissertation, STMIK AKAKOM Yogyakarta).
- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 6

Тема: Графічні зображення. Особливості веб-графіки. Частина 2

План заняття:

- 10. Можливості оптимізації*
- 11. Стилізація зображення*
- 12. Зміна розмірів зображення*
- 13. Фрагментна оптимізація*

Можливості оптимізації

Щоб будь-яке зображення можна було використовувати на Web-сторінці, його необхідно попередньо оптимізувати, зменшивши до мінімуму розмір файлу і, отже, час завантаження по мережі так, щоб якість зображення при цьому помітно не погіршився. Оптимізація це, по суті, компроміс між розміром файлу та якістю зображення.

У форматі JPEG зображення зберігається з глибиною кольору 24 біт і використовується стиснення з втратою інформації. У нього не так вже й багато способів оптимізації, точніше сказати, два:

- використання прогресивної розгортки;

- оптимальний коефіцієнт стиснення.

Використання прогресивної розгортки не позначається на розмірі зображення, але помітно при завантаженні. У разі запису зображення в стандартному форматі висновок на екран здійснюється горизонтальними смугами, і до завантаження всього зображення неможливо визначити його зміст. Навпаки, під час запису файлу в прогресивному форматі зображення з'являється відразу цілком, але в грубій формі, і потім поступово поліпшується. Це дає відвідувачам сайту можливість відразу ж оцінити зображення і вирішити, чи варто очікувати його кінцеве завантаження, що дозволяє заощадити трохи часу для перегляду сторінок. Загальний час завантаження сторінки при цьому не зменшується, виникає лише ілюзія його зменшення. Прогресивна розгортка JPEG підтримується всіма браузерами, але не кожен графічний пакет може записувати в цьому форматі. Останні версії Photoshop вміють це робити.

Як визначити оптимальний ступінь стиснення? Для цього потрібно зберегти зображення при декількох значеннях коефіцієнта стиснення, а потім уважно розглянути його. Той коефіцієнт стиску, при якому спотворення ще не видно, але зі збільшенням коефіцієнта на одну сходинку стають помітними, і буде оптимальним. Розмір файлу при такому коефіцієнті стиснення буде мінімально можливим при збереженні якості.

Формат GIF сильно відрізняється від JPEG і використовує зовсім інші методи оптимізації. Він призначений для зберігання зображень з кількістю кольорів не більше 256 і використовує стиснення без втрат за методом LZW, який застосовується в програмах-архіваторах. Методи оптимізації можна розділити на наступні типи, виключаючи оптимізацію файлів з анімацією:

- зменшення кількості кольорів;
- оптимізація палітри зображення;
- стилізація зображення;
- зміна розмірів зображення;

- фрагментарна оптимізація;
- оптимізація "прозорих" зображень.

Тепер розберемося по порядку з цими методами і прийомами оптимізації.

Зменшення кількості кольорів. Для більшості зображень, що не є фотографічними, кількість необхідних для відтворення кольорів часто менше 256. Зайві кольори можна прибрати з зображення, тим самим зменшивши його розмір. Це можливо тому, що формат GIF підтримує розмір палітри менше 256 кольорів. Теоретично ми можемо поставити будь-яке число кольорів палітри від 2 до 256, наприклад, 45 кольорів або 99. Але фактично кількість кольорів в зображенні вибирається з ряду: 2, 4, 8, 16, 32, 64, 128, 256. Це пов'язано з тим, що для представлення кольору використовується завжди ціле число біт, а наведений вище ряд є поруч максимальної кількості кольорів при використанні від 1 до 8 біт на піксель. У разі, якщо ви використовуєте в малюнку, припустимо, тільки 97 кольорів, для збереження одного пікселя все одно буде використано 7 біт, як і при використанні 128 кольорів. Тому при оптимізації кількості кольорів потрібно орієнтуватися на наведений вище ряд. Це дозволить отримати більш якісну картинку. Зменшивши кількість кольорів з 256 до 32, можна іноді домогтися зменшення розміру файлу в 2 рази без помітної втрати якості.

Оптимізація палітри зображення. Перетворення повнокольорового зображення в індексний, тобто зменшення кількості кольорів до 256 є важливою операцією, і тут слід сказати кілька слів про те, як це робиться, сучасними графічними пакетами.

Палітра кінцевого зображення може бути або фіксованою, або оптимізованою. У першому випадку графічний редактор переглядає кожну точку зображення і підбирає найближчу їй за кольором з палітри. Цей

спосіб дає найгірші результати з точки зору правильності відображення кольорів.

У другому випадку, коли використовується оптимізована палітра, графічний редактор спочатку аналізує малюнок і складає список усіх використовуваних в зображенні кольорів. Далі, на підставі частоти появи кольорів, складається палітра, яка називається оптимізованою. Після цього малюнок знову проглядається, і колір пікселя змінюється на найближчий з палітри. Цей спосіб дає набагато кращі результати, і саме його слід застосовувати, якщо у вас немає причин для використання фіксованого палітри.

При оптимізації конкретного зображення завдання полягає в тому, щоб вибрати оптимальну кількість кольорів у палітрі. Тут доречно сказати про спосіб збільшення кількості використовуваних кольорів при їх нестачі - так званому змішуванні (dithering). При цьому області, які в оригіналі були залиті однотонним кольором, після перетворення заповнюються сумішшю пікселів різних кольорів, розкиданих по випадковому закону. Як правило, кожен відсутній в скороченій палітрі колір передається пікселями двох найближчих до нього кольорів нової палітри. В результаті зображення набуває характерну зернисту, шорстку фактуру. Дуже часто змішування є єдиним способом хоч скільки-небудь адекватно передати вихідні кольору за допомогою палітри, в якій цих кольорів вже немає. Наприклад, у нас є 16 стандартних чистих кольорів, а нам потрібен відсутній помаранчевий колір. В такому випадку можна скласти його з червоних і жовтих точок, розмістивши їх в шаховому порядку. Здалеку буде здаватися, що це суцільний помаранчевий колір. Це класичний приклад змішування. Його використання при перетворенні зображень може дати дуже гарні результати. Але з точки зору оптимізації розмірів файлу відбувається зовсім протилежне. Змішування може збільшити розмір файлу, причому досить істотно - він може стати більше, ніж до оптимізації. А вся справа в способі

зберігання зображення форматом GIF. Зображення перед збереженням на диску піддається стисненню методом LZW, а основна особливість цього методу полягає в тому, що стисненню найкраще піддаються області, заповнені однорідним кольором, і найгірше - області, що складаються з набору різнокольорових точок.

Однак для дуже багатьох зображень, складових нерідко основу дизайну сторінки, змішування вкрай небажано. Що ж робити? Вихід, хоч і не найкращий, є. Виявляється, існує набір з 216 кольорів, які з гарантією ніколи не піддаються змішуванню ні в Netscape Navigator, ні в Internet Explorer, ні в Windows, ні в Macintosh. Якщо ваше зображення буде містити тільки кольори з цього набору або, принаймі, до цього набору будуть ставитися кольори, що займають найбільшу площу, в браузері воно буде виглядати набагато чистішим та привабливішим. Цю палітру часто називають безпечною (browser-safe palette).

Скористатися безпечною палітрою не пізно, навіть коли графіка вже готова. Однак, якщо ви хочете мати більший контроль над кольоровою стороною свого дизайну, ті графічні елементи, для яких хотілося б уникнути змішування, краще з самого початку малювати з використанням безпечної палітри. Безпечна палітра підтримується останніми версіями Adobe Photoshop.

Стилізація зображення.

Для зменшення кількості використовуваних кольорів в зображеннях, що містять фотографічні сюжети, можливе застосування стилізації фотографії. Для цього можна скористатися такими прийомами, як тонування зображення та застосування художніх фільтрів, імітуючих малюнки. Фотографії, оброблені таким способом, будуть стискуватися краще і вимагати палітру, що містить менше 256 кольорів.

Зміна розмірів зображення.

Цей метод оптимізації заснований на зміні розмірів малюнка. Його не можна застосовувати до звичайних малюнків, але він дуже добре підходить для малюнків, що складаються з елементів типу ліній і однорідних областей. Припустимо, нам знадобився малюнок, який представляє собою чорний квадрат розміром 100x100 пікселів. Браузер сам розтягне наш піксель до потрібних розмірів, і, таким чином, буде досягнута значна економія в обсязі файлу. Подібним же чином можна використовувати однопіксельні малюнки інших кольорів, але найбільшу користь може принести прозоре однопіксельне зображення GIF. З його допомогою зручно задавати «пустушки» на сторінках, абзацний відступи, фіксувати мінімальну ширину або висоту таблиці і робити інші корисні речі.

Фрагментарна оптимізація.

Фрагментарна оптимізація застосовується в випадку, якщо у вас є деяке велике зображення, що містить сильно риниться за кількістю кольорів області. У цьому випадку зображення можна розрізати на фрагменти і помістити їх в таблицю, тоді в браузері воно буде виглядати як єдине ціле. Тепер кожен фрагмент вихідного малюнка оптимізується окремо. Для фрагментів, що містять мало кольорів, цілком може вистачити палітри в 8 або 16 кольорів, а для багатих кольорів фрагментів можна використовувати повну палітру в 256 кольорів або зберігати їх у форматі JPEG. Цей спосіб дозволяє зменшити об'єм, займаний вихідним файлом, приблизно рази в 2-3, в залежності від конкретного зображення. Крім того, зображення буде з'являтися на екрані більш рівномірно, так як всі фрагменти будуть завантажуватися паралельно. Однак, якщо кількість фрагментів перевищить 10, то можливого виграшу не вийде, так як кожен фрагмент повинен буде зберігати свою копію палітри та службової інформації.

Оптимізація «прозорих» зображень. Формат GIF дозволяє зберігати так звані «прозорі» зображення. Насправді, GIF не підтримує прозорість в зображеннях - альфа-канал - він лише дозволяє призначити одному будь-

якого кольору в палітрі атрибут прозорості. Якщо браузер малює на екрані таке зображення, то, зустрічаючи "прозорий" піксель, він просто ігнорує його і не показує на екрані. Само по собі це добре, але в разі використання методу згладжування країв зображення (anti-aliasing) і наступного збереження в форматі GIF, виникає проблема появи ореолу навколо прозорого зображення у випадку, якщо GIF показується на тлі, відмінному від того, на якому відбувалося згладжування і який став прозорим. Єдиним способом зменшення прояву цього ефекту є призначення прозорості кольору, близького до кольору фону, на якому буде використовуватися GIF, і проведення згладжування на цьому ж тлі.

Який формат використовувати? Питання про те, в якому форматі зберегти потрібне вам зображення, не такий простий, як здається. Дуже часто можна прочитати, що фотографічні зображення, оскільки вони мають більше 256 кольорів, слід зберегти в форматі JPEG, а малюнки в форматі GIF. Це правило дійсно вірно в більшості випадків. Однак можуть бути і винятки. У ряді випадків, якщо вам потрібна особлива якість зображення, слід віддати перевагу GIF. Справа в тому, що головною похибкою у використанні JPEG є те, що зображення, збережені в цьому форматі, на відміну від GIF мають візуально помітні спотворення. Вважається, що якщо вихідний розмір файлу стискається до 10 разів то спотворення будуть не дуже помітні. У той же час, якщо в досить «інтелектуальною» програмою зменшити кількість кольорів фотографії до 256 цих 256 кольорів в більшості випадків вистачить на те, щоб зображення ж прогало в візуальному сприйнятті.

У всякому разі, майте на увазі наступні обставини:

- Чим більше на малюнку дрібних і контрастних деталей, тим сильніше будуть помітні спотворення, що вносяться при стисканні в форматі JPEG.

- Чим більше геометричні розміри зображення в пікселях і чим нижче контрастність знімка, тим спотворення JPEG будуть менше помітні.

- Чим більше на малюнку контрастних і дрібних деталей, тим більший розмір файлу вийде при збереженні в форматі GIF.

- Чим більшу площу займають однорідно забарвлені ділянки і чим більшу горизонтальну протяжність мають деталі малюнка, тим менший розмір займе на диску файл GIF.

Але є дві області, де GIF - поза конкуренцією. Перша - це створення анімованих, тобто рухомих зображень. Насправді анімований GIF-це кілька графічних зображень, записаних в один файл. Послідовний показ цих зображень створює ілюзію руху. У файлі GIF зберігається інформація про те, скільки разів в секунду треба міняти зображення, в якій послідовності показувати картинки і т.д. Для створення анімованого зображення ви в будь-якому графічному редакторі малюєте картинку, з яких буде будуватися анімація, а потім завантажуєте їх в спеціальну програму підготовки анімації, де вказуєте, як ці картинки повинні чергуватися. Таких програм існує велика кількість.

Інша незамінна властивість GIF - це можливість отримання прозорого фону зображення, тобто такого фону, крізь який видно все, що під ним знаходиться. Створювати GIF з прозорим фоном може будь-який професійний графічний редактор.

Питання для самозакріплення:

- 4) Які види графічних форматів вам відомі?
- 5) Назвіть переваги GIF
- 6) Назвіть способи оптимізації графічних файлів.

Рекомендована література:

- Wahyudin, A. (2019). IMPLEMENTASI RESPONSIVE WEB DESIGN PADA SISTEM INFORMASI GEOGRAFIS OBJEK CENTRAL BIMA (NTB) (Doctoral dissertation, STMIK AKAKOM Yogyakarta).
- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 7

Тема: Структура HTML документу

План заняття:

14. Основні поняття HTML

15. Структура HTML документу

16. Специфікація елемента BODY

Основні поняття HTML

HTML - мова гіпертекстової розмітки документів (HyperText Markup Language). За допомогою HTML створюються Web-сторінки, які знаходяться в глобальній комп'ютерній мережі Інтернет. HTML - це не мова програмування в традиційному сенсі, вона є мовою розмітки. За допомогою HTML текстовий документ розбивають на блоки смислової інформації (заголовки, параграфи, таблиці, малюнки тощо).

Гіпертекстовий документ - це документ, що містить переходи (**гіперпосилання**) на інші документи. При використанні гіперпосилання відбувається переміщення від одного документа до іншого (як по ланцюжку) в Інтернеті.

Особливості HTML-документа:

1. HTML-документ може містити текст, графіку, відео і звук.
2. У загальному випадку HTML-документ - це один або кілька текстових файлів, що мають розширення. htm або html.

3. Створювати HTML-документ можна як за допомогою спеціальних програм - редакторів HTML, так і за допомогою будь-якого текстового редактора.

4. Для перегляду HTML -документів існують спеціальні **програми-браузери**. вони **інтерпретують** HTML-документи, тобто переводять текст документа в Web-сторінки, і відображають її на екрані користувача.

5. Якщо при інтерпретації HTML-документа браузер чогось не розуміє, то повідомлення про помилку не виникає, а це місце в HTML-документі ігнорується і не відображається браузером.

HTML-документ складається з елементів HTML.

Елемент HTML - це найчастіше два тега (відкривачий і закриваючий) і частина документа між ними. Крім того, існують елементи HTML, що складаються тільки з одного тега.

Тег - в перекладі з англійської - ярлик, етикетка. Тег визначає тип виведеного елемента HTML (наприклад, заголовок, таблиця, малюнок і т.п.). Сам тег не відображається браузером. Тег є послідовність елементів:

- символ лівої кутової дужки (<) - початок тега;
- необов'язковий символ слеша (/) - символ використовується, щоб позначити закриваючий тег;
- ім'я тега;
- необов'язкові атрибути в відкриваючому тезі;
- символ правої кутової дужки (>)

Атрибути - необов'язковий набір параметрів, що визначають додаткові властивості елемента HTML (наприклад, колір або розмір). Атрибут складається:

- з назви ознаки;
- знака рівності (=);
- значення атрибута - рядки символів, укладеної в лапки

Приклад елемента HTML:

```
<H 1 ALIGN = "CENTER"> Глава 1 </ H 1>
```

У цьому прикладі:

<H 1 ALIGN = "CENTER"> - відкриває тег

</ H 1> - закриває тег

H 1 - ім'я тега

ALIGN = "CENTER" - атрибут

ALIGN - ім'я атрибута

"CENTER" - значення атрибута

Правила створення HTML -документів:

1. Теги та атрибути можна записувати в будь-якому регістрі, тобто </ H 1> і </ h 1> - це одне і те ж.

2. Кілька прогалін поспіль, символи табуляції і переведення рядка при інтерпретації браузером замінюються на один пробіл. Це дозволяє писати добре структуровані вихідні тексти файлів HTML.

СТРУКТУРА HTML-документа

Кожен HTML-документ повинен починатися тегом **<HTML>** і закінчуватися тегом **</ HTML>**. Ці теги позначають, що знаходяться між ними рядка представляють єдиний HTML-документ. Крім того, можна помітити, що файл HTML в цілому є елементом мови HTML.

Також в HTML-документи повинні бути присутніми елементи **HEAD** (інформація про документ) і **BODY** (тіло документа).

Розділ документа **HEAD** визначає його заголовок, а також містить додаткову інформацію про документ для браузера.

Розділ заголовка починається тегом **<HEAD>** і слідує відразу за тегом **<HTML>**.

Для того щоб дати назву HTML-документа, призначений тег **<TITLE>**. Цю назва буде виведено в заголовок вікна браузера. Назва записується між тегами **<TITLE>** і **</ TITLE>** і являє собою рядок тексту. Довжина цього рядка може бути будь-яка, але рекомендується робити її не більш 60 символів. Елемент **TITLE** повинен знаходитися тільки в розділі **HEAD**.

Розділ **BODY** повинен починатися тегом **<BODY>** і завершуватися тегом **</ BODY>**, між якими розташовуються елементи HTML, з яких і складається зміст документа.

Специфікація елемента **BODY**

Тег **<BODY>** має ряд атрибутів, що визначають зовнішній вигляд документа. Нижче наводиться специфікація тега **<BODY>**.

```
<BODY  
TEXT = "колір тексту"  
BGCOLOR = "колір фону"  
BACKGROUND = "адреса фонового малюнка"  
LINK = "колір невідвідування гіперпосилання"  
VLINK = "колір відвідується гіперпосилання"  
ALINK = "колір активного гіперпосилання"  
>
```

Атрибут **TEXT** задає колір шрифту для всього документа в форматі **RGB** або в символній нотації. За замовчуванням (якщо не вказано цей атрибут) буде використано стандартні браузера.

Атрибут **BGCOLOR** задає колір фону вікна браузера документа в форматі **RGB** або в символній нотації. За замовчуванням використовуються настройки браузера.

Атрибут **BACKGROUND** дозволяє вказати адресу і ім'я малюнка, який використовується в якості фону. Цей малюнок буде розмножений і розподілений на задньому плані документа.

Атрибути **LINK**, **VLINK** і **ALINK** задають кольори гіперпосилань в форматі **RGB** або в символній нотації. За замовчуванням використовуються настройки браузера. Невідвідування гіперпосилання -

гіперпосилання, яка ще не використовувалася для переходу до іншого документу. Відвідана гіперпосилання - гіперпосилання, яка вже використовувалася для переходу до іншого документу. Активне гіперпосилання - гіперпосилання на документ, до якого в даний момент відбувається перехід.

Поради щодо використання атрибутів тега BODY

- Якщо ви вказуєте хоча б один колір в тегу BODY, то вкажіть і інші. Це пов'язано з тим, що користувач може встановити настройки кольорів свого браузера як йому зручніше. Вказівка тільки одного кольору може привести до ситуації, що у деяких користувачів текст зіллється з кольором фону. В результаті перегляд документа буде ускладнений.

- Вибирайте колір тексту так, щоб він "працював" разом з кольором фону або основними кольорами зображення. Наприклад, червоне на зеленому може викликати серйозні проблеми у значної кількості людей.

- В елементі BODY можна задати як BGCOLOR, так і BACKGROUND. В цьому випадку браузер віддає перевагу BACKGROUND, але якщо зображення фону неможливо завантажити, буде використано BGCOLOR. Тому намагайтеся задавати колір фону схожим на колір шпалер, щоб не порушився баланс кольорів документа.

Приклад простого HTML-документа

```
<HTML>
<HEAD>
  <TITLE> Привітання </TITLE>
</HEAD>
<BODY>
  <P> Ласкаво просимо! </P>
</BODY>
</HTML>
```

Цей документ відобразиться в браузері так:

Ласкаво просимо!

Рис. 7.1. Відображення тексту

Приклад використання фонового малюнка

```
<HTML>  
<HEAD>  
  <TITLE>Привітання</TITLE>  
</HEAD>  
<BODY BACKGROUND = "smile.png">  
  <P> Ласкаво просимо! </P>  
</BODY>  
</HTML>
```

Цей документ відобразиться в браузері так:

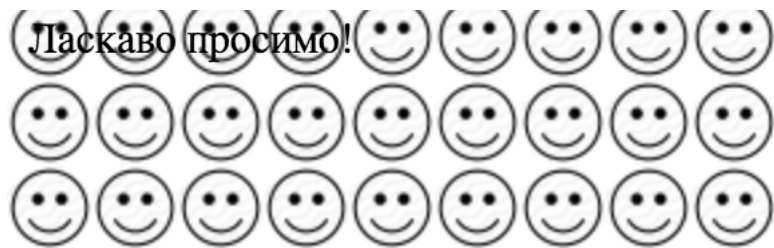


Рис. 7.2. Відображення тексту

1. Питання для закріплення.

- 1) Дайте поняття елементу HTML, тега, атрибутів.
- 2) Що таке браузер?
- 3) Які особливості та правила HTML-документа?
- 4) Які елементи HTML входять в обов'язкову структуру HTML-документа?
- 5) Для чого використовується елемент BODY, і які він має атрибути?

Рекомендована література:

- Wahyudin, A. (2019). IMPLEMENTASI RESPONSIVE WEB DESIGN PADA SISTEM INFORMASI GEOGRAFIS OBJEK CENTRAL BIMA (NTB) (Doctoral dissertation, STMIK AKAKOM Yogyakarta).

- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 8

Тема: Кольорові специфікації. Графічні елементи

План заняття:

17.Колірні специфікації

18.Графічні елементи

КОЛІРНІ СПЕЦИФІКАЦІЇ

Для визначення кольору різних елементів HTML-документа необхідно вказати значення відповідних атрибутів. Вказувати значення цих атрибутів можна двома способами:

- визначати колір в форматі RGB;
- визначати колір, використовуючи символну нотацію

Формат RGB - це система кольору, яка базується на змішуванні трьох основних кольорів: червоному (RED), зеленому (GREEN) і синьому (BLUE). Підсумковий колір визначається цифрами в шістнадцятковому коді. Для кожного кольору задається шістнадцяткове значення в межах від 0 до FF, що відповідає діапазону 0-255 в десятковому численні. Потім ці значення об'єднуються в одне число, перед яким ставиться символ #. Наприклад, число #800080 позначає фіолетовий колір. Вказуючи колір в форматі RGB, можна визначити більше шістнадцяти мільйонів кольорних відтінків.

СИМВОЛЬНА НОТАЦІЯ

Завдання кольору в форматі RGB має один недолік - необхідно пам'ятати сукупності цифр для вказівки кольору. Цього недоліку позбавлена символна нотація. Можна вказувати назву кольору

англійською мовою. Але у цього способу вказівки кольору теж є недолік - визначено всього шістнадцять імен квітів.

ВІДПОВІДНІСТЬ ФОРМАТУ RGB І СИМВОЛЬНОЇ НОТАЦІЇ

Нижче наведена таблиця відповідностей вказівки кольору в символній нотації і форматі RGB.

символьна нотація	формат RGB	колір
Black	# 000000	чорний
Silver	# C0C0C0	срібло
Gray	# 808080	сірий
White	#FFFFFF	білий
Maroon	# 800000	темно-бордовий
Red	# FF0000	червоний
Purple	# 800080	фіолетовий
Fuchsia	# FF00FF	рожевий
Green	# 008000	зелений
Lime	# 00FF00	вапно
Olive	# 808000	оливковий
Yellow	# FFFF00	лимонний
Navy	# 000080	Темно синій
Blue	# 0000FF	синій
Teal	# 008080	Темно-бірюзовий
Aqua	# 00FFFF	бірюзовий

Таким чином, рядок TEXT = "#008000" і рядок TEXT = "Green" в тезі <BODY> однаково визначають колір шрифту - зелений.

ГРАФІЧНІ ЕЛЕМЕНТИ

Одним з достоїнств HTML-документа є можливість використання графічних елементів в оформленні. Можна виділити три елементи, найчастіше використовуваних в HTML-документа: горизонтальні лінії, таблиці і малюнки.

ГОРИЗОНТАЛЬНІ ЛІНІЇ

Горизонтальні лінії візуально підкреслюють закінченість тій чи іншій області документа. Зараз часто використовують рельєфну, вдавлену лінію, щоб позначити "об'ємність" документа.

Тег <HR> дозволяє провести рельєфну горизонтальну лінію у вікні більшості браузерів. Цей тег не є контейнером, тому не вимагає закриття тега. До і після лінії автоматично вставляється порожній рядок. Специфікація тега <HR>:

<HR

ALIGN = "вирівнювання"

WIDTH = "довжина лінії"

SIZE = "товщина лінії"

NOSHADE

>

Атрибут ALIGN визначає спосіб вирівнювання лінії. Він може мати наступні значення:

- LEFT - лінія вирівнюється по лівому краю вікна браузера. Це значення використовується за умовчанням.

- CENTER - лінія вирівнюється по центру вікна браузера.

- RIGHT - лінія вирівнюється по правому краю вікна браузера.

Атрибут WIDTH задає довжину лінії. Значенням даного атрибута є число. Це число означає довжину лінії в пікселях. Якщо після числа варто знак%, то це означає довжину в процентах від ширини вікна. наприклад:

<HR WIDTH = "400"> - лінія довжиною 400 пікселів.

<HR WIDTH = "50%"> - лінія довжиною 50 відсотків від ширини вікна.

Атрибут SIZE задає товщину лінії. Значення цього атрибута є число. Це число означає товщину лінії в пікселях.

Атрибут NOSHADE скасовує "тривимірність" лінії.

МАЛЮНКИ

Без ілюстрацій документ нудний, млявий і одноманітний. HTML дозволяє використовувати малюнки в форматі JPG і GIF для оформлення HTML-документів. Для вставки малюнків використовується тег . Специфікація тега :

```
<IMG
```

```
SRC = "адреса малюнка"
```

```
ALIGN = "вирівнювання"
```

```
HEIGHT = "висота малюнка"
```

```
WIDTH = "ширина малюнка"
```

```
BORDER = "товщина рамки"
```

```
>
```

Атрибут SRC визначає URL-адресу малюнка, який буде відображатися браузером.

Атрибут ALIGN визначає спосіб вирівнювання малюнка. Він може мати наступні значення:

- TOP - малюнок вирівнюється по верхньому краю поточного рядка.

- MIDDLE - малюнок вирівнюється серединою по поточному рядку.

- BOTTOM - малюнок вирівнюється по нижньому краю поточного рядка.

- LEFT - малюнок притискається до лівого краю вікна браузера і обтікається текстом.

- RIGHT - малюнок притискається до правого краю вікна браузера і обтікається текстом.

Атрибут HEIGHT визначає висоту малюнка в пікселях.

Атрибут WIDTH визначає ширину малюнка в пікселях.


Використовуючи атрибути HEIGHT і WIDTH можна збільшувати або зменшувати малюнок. Якщо вказати тільки один з цих атрибутів, то малюнок буде збільшений або зменшений пропорційно і по ширині, і по висоті.


Атрибут BORDER дозволяє задавати рамку навколо малюнка. Значення цього атрибута - товщина рамки в пікселях. За замовчуванням - 1.

Приклад вирівнювання малюнків:

```
<HTML>
<HEAD>
<TITLE>Приклад вирівнювання</TITLE>
</HEAD>
<BODY>
<P>Вирівнювання<IMG SRC="ris1.jpg" ALIGN="TOP">по
верхньому краю</P>
<P>Вирівнювання<IMG SRC="ris1.jpg" ALIGN="BOTTOM">
по нижньому краю</P>
<P>Вирівнювання<IMG SRC="ris1.jpg" ALIGN="MIDDLE">
по середині</P>
</BODY>
</HTML>
```

Браузер відобразить даний приклад так:

Вирівнювання  по верхньому краю

Вирівнювання  по нижньому краю


Вирівнювання  по середині

Рис. 8.1. Приклад вирівнювання тексту відносно малюнків

2. Питання для закріплення.

- 1) Чим відрізняється символна нотація від формату RGB?
- 2) Які значення в символній нотації можна використовувати для вказівки кольору?

Рекомендована література:

- Wahyudin, A. (2019). IMPLEMENTASI RESPONSIVE WEB DESIGN PADA SISTEM INFORMASI GEOGRAFIS OBJEK CENTRAL BIMA (NTB) (Doctoral dissertation, STMIK AKAKOM Yogyakarta).
- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 9

Тема: Гіперпосилання

План заняття:

- 19. Основні поняття*
- 20. Універсальний ідентифікатор ресурсів URL*
- 21. Внутрішні посилання*
- 22. Посилання на документи різних типів*

Гіперпосилання - переходи до інших документів. Вони є основою HTML. Гіперпосилання можна використовувати для переходу не тільки до інших HTML-документів, але і іншим об'єктам, які можна розмістити на комп'ютері, наприклад, до відео- та аудіофайлів, архівів, малюнків і т.п.

Кожне посилання складається з двох частин. Перша - це те, що відображається браузером. Вона називається покажчиком посилання. Друга частина - адресна, що містить адресу об'єкта, до якого повинен відбуватися перехід. Цю частину називають URL (універсальний ідентифікатор ресурсів).

Коли ви натискаєте мишею по покажчику посилання, браузер завантажує документ, адреса якого вказана в URL.

Покажчиком посилання може бути слово, група слів або малюнок. Якщо покажчик текстовий, то він зазвичай відображається браузером підкресленим синім шрифтом. При наведенні курсора миші на покажчик курсор приймає форму руки, вказуючи, що це посилання і можна зробити перехід. Якщо покажчик графічний, зовні він не відрізняється від інших малюнків, але при наведенні курсору миші на такий покажчик, курсор також приймає форму руки.

Універсальний ідентифікатор ресурсів URL

За своєю суттю URL - це адреса файлу, до якого відбувається перехід. Вказівка адреси може бути відносною або абсолютною.

Коли в URL вказується повний шлях до файлу, незалежно від того, де цей файл знаходиться (в Інтернеті або на жорсткому диску комп'ютера), то це абсолютна вказівка. Наприклад: <http://www.mysite.ua/page.htm> - абсолютна адреса документа, що знаходиться в Інтернеті, `c:\web\doc1.htm` - абсолютний адреса документа на диску c.

Якщо в URL вказується не повний шлях, а шлях щодо адреси документа, в якому знаходиться посилання, то це відносна вказівка. Наприклад, браузер відображає документ, абсолютна адреса якого `c:\web\doc1.htm`, в цьому документі є посилання з адресою `pic1 / ris1.jpg`, то це означає, що насправді посилання буде на документ за адресою `c:\web\pic1\ris1.jpg`.

Коли треба послатися на файл, що знаходиться на іншому комп'ютері, тоді слід користуватися абсолютним зазначенням, а якщо посилатися на файли, що знаходяться на тому ж комп'ютері, де і документ, що містить посилання, то краще використовувати відносну вказівку.

Правила запису лінків

Для організації посилання необхідно вказати браузеру, що є покажчиком посилання, а також визначити адресу документа, на який відбувається посилання. Обидві ці дії виконуються з допомогою тега `<A>`.

Тег `<A>` має наступну специфікацію:

```
<A  
HREF = "URL-адресу"  
NAME = "ім'я посилання"  
TARGET = "об'єкт для виведення"  
>
```

Атрибут `HREF` використовується для завдання адреси файлу, до якого проводиться перехід. Значення цього атрибута є текстовий рядок, що містить абсолютний або відносний URL-адресу.

Атрибут `NAME` призначений для завдання посиланню імені. Значення цього атрибута є коротка текстова рядок. Цей атрибут використовується для посилань усередині одного HTML-документа.

Атрибут `TARGET` дозволяє визначити, куди буде виводитися документ, на який відбувається перехід. Цей атрибут може мати значення `_blank` - це означає, що документ буде виводитися в новому вікні браузера.

Приклад посилання:

```
<A HREF="doc1.htm"> Документ 1 </A>
```

Браузер відобразить цей рядок так:

Документ 1

При натисканні мишею на цьому рядку браузер завантажить і відобразить файл doc1.htm, тобто "Документ 1" - це покажчик посилання, а "doc1.htm" - URL-адресу.

Внутрішні посилання

Крім посилань на інші документи, часто буває корисно включити посилання на різні частини поточного документа. Наприклад, великий документ читається краще, якщо він має зміст з посиланнями на відповідні розділи.

Для побудови внутрішньої посилання спочатку потрібно створити покажчик, який визначає місце призначення. Для цього в місці, куди потім буде проводитися посилання, треба помістити тег <A> з атрибутом NAME, визначивши цим атрибутом ім'я покажчика. наприклад:

```
<A NAME="glava5"> </A>
```

Зверніть увагу, що в цьому прикладі відсутня зміст тега <A>. Зазвичай так і роблять, оскільки тут немає необхідності якимось виділяти текст, а потрібно лише вказати місце розташування.

Після того як місце призначення визначено, можна приступити до створення посилання на нього. Для цього в атрибуті HREF тега <A> поміщається ім'я посилання з префіксом #, що говорить про те, що це внутрішнє посилання.

наприклад:

```
<A HREF="#glava5"> Глава 5 </A>
```

Тепер, якщо користувач клацне кнопкою миші на словах "Глава 5", то браузер виведе відповідну частину документа у вікні перегляду.

Можна поєднувати внутрішні посилання з посиланнями на інші документи. наприклад:

` Глава 5 Документа 1 `

При натисканні на це посилання браузер відкриє файл doc1.htm, знайде в цьому файлі покажчик glava5 і виведе у вікні перегляду відповідну інформацію.

Посилання на документи різних типів

Коли користувач клацає мишею на посиланні, що вказує на інший HTML-документ, то документ виводиться безпосередньо у вікні браузера. Якщо ж посилання вказує на документ іншого типу, програма перегляду приймає документ і потім вирішує, що з ним робити далі.

Браузер знає цей тип документа і вміє з ним працювати. Наприклад, якщо вставити посилання на графічний файл формату GIF і клацнути мишею на ньому, то браузер очистить вікно перегляду і відобразить вказане зображення.

Браузер не розпізнає тип прийнятого документа і не знає, що з ним робити далі. У цьому випадку він звернеться до допоміжних програм, розміщених на комп'ютері користувача. Якщо відповідна програма знайдеться, браузер запустить її і передасть їй отриманий документ для обробки. Наприклад, якщо користувач клацне на посиланні, що містить відеофайл формату AVI, браузер завантажить файл, знайде програму для демонстрації AVI-файлів і запустить її. Відеофайл буде показаний в додатковому невеликому вікні.

Посилання на ресурси Інтернету

Основне призначення HTML-документів - це глобальна комп'ютерна мережа Інтернет. HTML-документ, розміщений в Інтернеті, стає Web-сторінкою. Щоб звернутися до Web-сторінці, треба вказати URL-адресу в такій формі: `http://sitename/docname`, де sitename - ім'я сайту, docname - ім'я документа. Наприклад: `http://kotoz.newmail.ua/autor.htm`.

Можна на Web-сторінці розмістити адресу електронної пошти. Для цього URL-адресу вказується так: mailto: address, де address - це адреса поштової скриньки. Наприклад: mailto: admin@ldubgd.edu.ua.

Також існують спеціальні формати URL-адреси для інших ресурсів Інтернету (FTP, TelNet, Newsgroup, Gopher, WAIS).

3. Питання для закріплення.

- 1) Що таке URL?
- 2) Що таке внутрішнє помилання та як воно створюється?
- 3) На що можна посилатися в HTML документах?

Рекомендована література:

- Wahyudin, A. (2019). IMPLEMENTASI RESPONSIVE WEB DESIGN PADA SISTEM INFORMASI GEOGRAFIS OBJEK CENTRAL BIMA (NTB) (Doctoral dissertation, STMIK AKAKOM Yogyakarta).
- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та
телекомунікацій

Основи графічного дизайну та проектної графіки

Лекція 10

Тема: Основні принципи роботи з Dreamweaver

План заняття:

23. Загальні відомості

24. Принцип роботи

Dreamweaver MX - захоплююча програма для створення Web-сторінок. Її можливості ширше в порівнянні зі звичайними функціями подібних редакторів. Dreamweaver дозволяє створювати об'єкти і сторінки з використанням мови HTML з усіма його формами, фреймами та іншим. Підтримує каскадні таблиці стилів, шари, об'єкти і можливість вставки інтерактивних елементів JavaScript і ActiveX, аплетів дії JavaScript. Також додавання анімації Macromedia Flash і Shockwave

Розробнику Web-проекту пропонуються гнучкий і потужний інструментарій управління сайтом, включаючи, вбудований повноцінний FTP-клієнт, візуальні карти сайтів і контроль над посиланнями. Можливості

Dreamweaver MX значно полегшують процес створення сторінок і управління сайтом.

Запуск редактора

Для запуску редактора Dreamweaver MX 2004 необхідно виконати наступну послідовність дій:

Пуск – Програми – Macromedia – Macromedia Dreamweaver MX.

Вибір інтерфейсу

При першому запуску Dreamweaver MX пропонується два варіанти виду робочої області редактора:

- Designer
- Coder

Чим вони відрізняються - зрозуміло з їх назв. Це вікно з'являється, як правило, тільки при першому завантаженні редактора. Надалі налаштування можна буде змінити командою меню Edit • Preferences (Редагувати • Уподобання). У діалоговому вікні Preferences (Уподобання) в списку Category (Категорія) виберіть пункт General (Загальні) і натисніть кнопку Change Workspace (Зміна робочої області). Після цього відкриється вікно Workspace Setup (Установка робочої області).

Відразу слід вибрати дизайнерський вигляд робочої області редактора. Так буде простіше працювати. Після того як вибір зроблений, завантаження редактора продовжиться. Після закінчення завантаження на екрані з'явиться робоча область редактора. В її центрі розміщується так зване стартове вікно редактора.

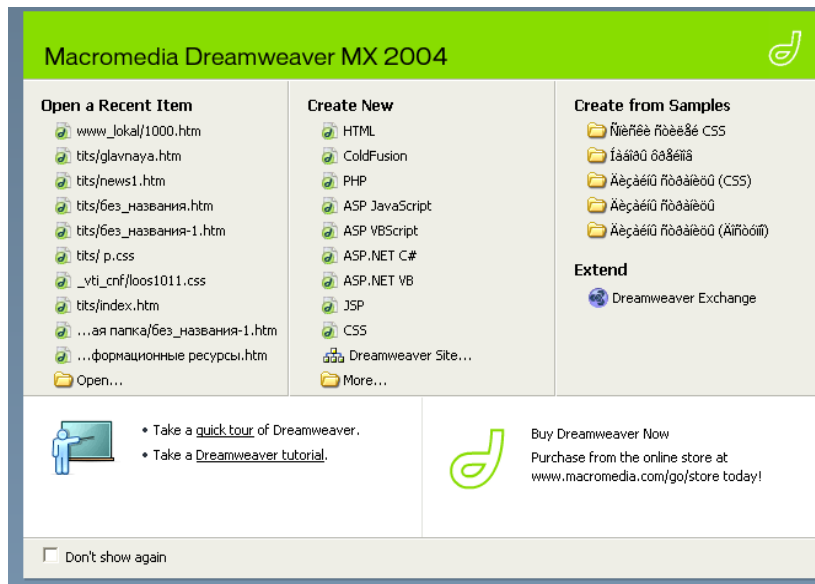


Рис. 10.1 – Стартове вікно редактора

Визначення та налаштування сайту

Визначення (або реєстрацію) нового сайту в Dreamweaver MX рекомендується проводити ще на початковому етапі побудови веб-сторінок, які входять в нього. По-перше, так буде зручніше потім, це дозволить уникнути плутанини при сортуванні веб-сторінок. По-друге, деякі дуже корисні функції редактора будуть доступні тільки після визначення сайту. І нарешті, на етапі опублікування сайту в Інтернеті не виникне проблем з адресацією посилань і демонстрацією малюнків. А ще при використанні на різних веб-сторінках одних і тих же малюнків (наприклад, логотипу фірми або значка лічильника відвідувань) знадобиться не безліч копій таких малюнків для кожної сторінки, а всього лише один малюнок.

Отже, для початку визначення нового сайту виконуємо команду меню Site • Manage Sites (Сайт • Управління сайтами). Ця команда меню відкриває діалогове вікно Manage Sites (Управління сайтами). Діалогове вікно, яке називається менеджером сайтів, відображає всі вже наявні проекти (сайти), а також дає можливість визначити новий сайт. Для цього в колонці кнопок у правій межі вікна клацаємо по самій верхній кнопці New (Новий).

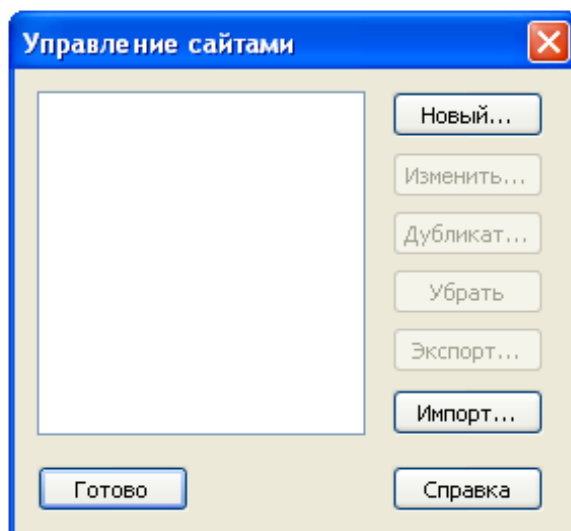


Рис. 10.2 – Діалогове вікно програми

У списку слід вибрати Site (Сайт). Dreamweaver відразу ж позначить ім'я нового сайту як Unnamed Site (Безіменний сайт). Відкриється нове діалогове вікно Site Definition (Визначення сайту). Саме тут і проводиться визначення сайту. Визначення сайту можна проводити двома способами - автоматично і вручну. Цим способам відповідають дві вкладки вікна - Basic (Основний) і Advanced (Розширений). Визначення сайту по першій вкладці - Basic - є найбільш простим. У покроковому режимі вам потрібно відповідати на питання чи вводити необхідні дані. Тут визначаються лише найосновніші, необхідні настройки сайту. Зрозуміло, що такий режим буде зручний початківцям.

Другий режим визначення сайту - Advanced - є, як випливає з назви, розширеним. Він більше підходить професіоналам, так як містить більш тонкі налаштування сайту. Тут користувач вже сам вибирає, які йому робити установки і в якій послідовності. Однак в будь-який момент часу ви можете перейти з одного режиму в інший.

Тепер розглянемо, які установки необхідно зробити при визначенні сайту. Звичайно, по-перше, необхідно вказати ім'я сайту. У режимі визначення сайту Basic визначення імені сайту є першим кроком. Для першого кроку на вкладці Basic є всього одне поле введення - це поле

введення імені сайту. Слід зауважити, що не варто давати сайту занадто довге ім'я, а писати краще латинськими буквами. Адже це ім'я - лише умовне ім'я тієї папки, в яку будуть записуватися всі файли, які використовуються в даному проекті, і ніхто, крім вас, цього імені не знатиме. Для прикладу ми назвемо сайт Site 1. Тепер натискаємо кнопку Next (Далі) у правому нижньому кутку вікна і переходимо до наступного кроку. На другому кроці пропонується визначити, чи будемо ми використовувати сервісні технології чи ні. А поки відмовляємося від їх використання і переходимо далі.

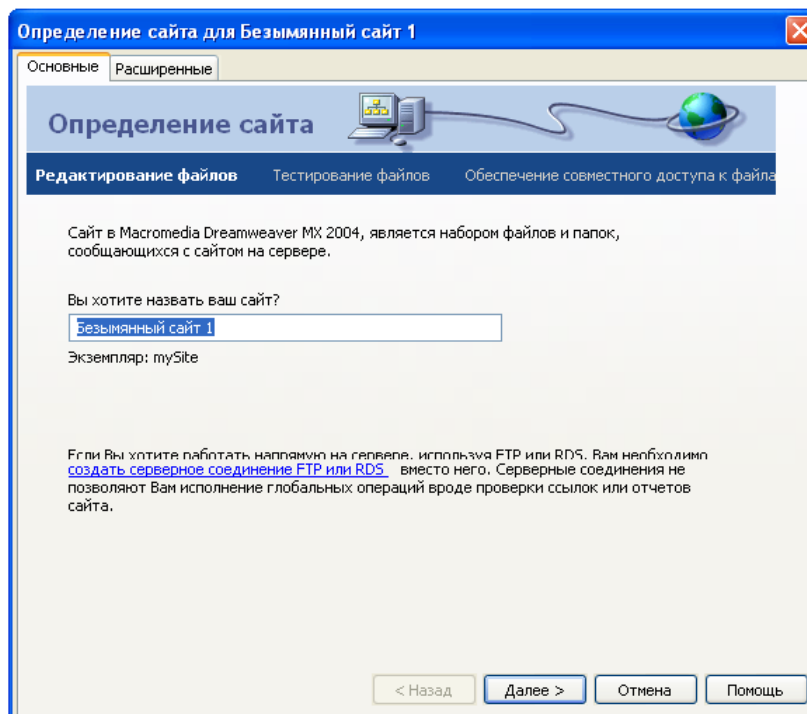


Рис. 10.3 – Визначення сайту для «Безіменного сайту 1»

Третій крок є одним з найважливіших після визначення імені сайту. Тут ми вибираємо, чи будемо створювати сайт на своєму комп'ютері (Edit local copies on my machine, then upload to server when ready), або на сервері по локальній мережі (Edit directly on server using local network). Як і рекомендує Dreamweaver, ми вибираємо перший варіант. В наступному полі вказуємо папку (каталог), в якій будуть зберігатися всі документи нашого сайту. Зручніше, коли ім'я сайту збігається з ім'ям папки. Якщо

такий папки ще немає, створюємо її тут же. Слід особливо наголосити на важливості цього кроку в ситуаціях, коли планується створення багатосторінкового сайту або вставка на веб-сторінку малюнків і гіперпосилань. Визначення кореневої папки сайту необхідно для створення відносних посилань (тобто посилань, не прив'язаних до конкретного комп'ютера), щоб ці посилання також працювали і на віддаленому сервері.

Знову натискаємо кнопку Next для переходу до четвертого кроку. На четвертому кроці вказуємо, яким способом будемо проводити закачування файлів нашого сайту на сервер. Ми вибрали найтипівіший для цього випадку протокол FTP, вказали адресу провайдера, який надає нам місце для сайту, а також реєстраційне ім'я (логін) і пароль для виходу в Інтернет. Втім, цей крок важливий тільки в тому випадку, якщо передбачається закачування сайту на сервер за допомогою Dreamweaver MX 2004, а виконати його можна і безпосередньо перед самою закачуванням. Тут же, клацнувши на кнопці Test Connection (Тест з'єднання), можна перевірити працездатність вказаних нами даних. На заключному етапі Dreamweaver MX 2004 підсумує всі введені дані і покаже їх вам. На будь-якому з кроків ви завжди можете повернутися назад і що-небудь виправити. Режим визначення сайту Advanced є більш професійним і дозволяє робити спеціалізовані установки. Цей режим дещо відрізняється від щойно розглянутого. У лівій межі вікна розташований список Category. Перераховані в ньому категорії по суті є назвами кроків в режимі Basic (основні). Вибираємо потрібну категорію і виробляємо необхідні установки. За замовчуванням відкрита категорія Local Info (Розширені). Тут вказується ім'я сайту (Site Name), локальна коренева папка (Local root folder). Крім того, можна включити автоматичне оновлення сайту на сервері.

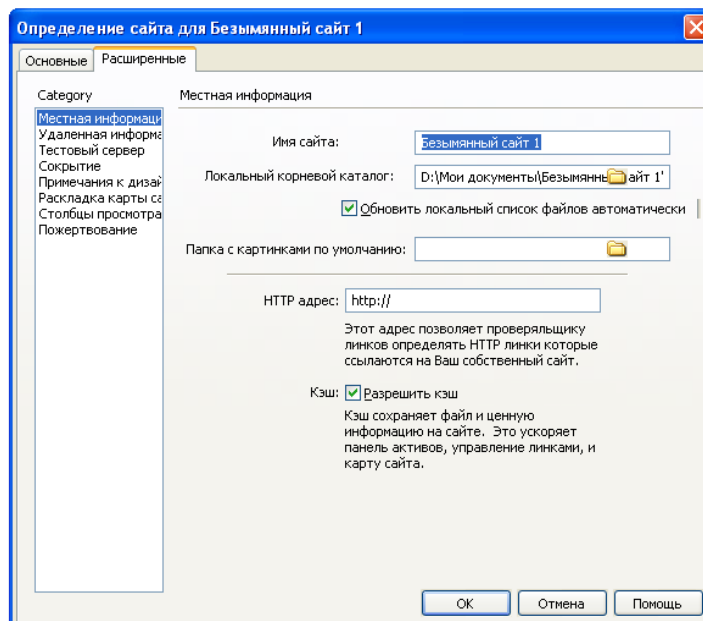


Рис. 10.4 – Налаштування сайту, місцева інформація

Наступна категорія - Remote Info (Дистанційна інформація) - задає параметри копії сайту на віддаленому сервері. Тут дуже важливо визначити режим завантаження файлів сайту на сервер (Access). Від цього будуть залежати всі інші настройки даної категорії. У разі вибору параметра завантаження по FTP, то всі наступні настройки будуть пов'язані саме з цим протоколом. Так, далі в поле FTP host вводимо адресу FTP-сервера, наприклад ftp://ftp.narod.ua. Ця електронна адреса можна дізнатися у провайдера, який надає вам місце на сервері для сайту. Нижче вводяться ім'я кореневої директорії сайту - Host directory, ім'я користувача і пароль, під якими здійснюється вхід в мережу. А поруч розташована кнопка Test (Тест), натиснувши на яку ви зможете тут же перевірити працездатність введених даних. Якщо FTP-сервер, за допомогою якого проводиться копіювання файлів сайту на сервер, підтримує тільки пасивний протокол FTP, то слід встановити прапорець Use passive FTP (Використовувати пасивний FTP). Якщо ви використовуєте проксі-сервер, то слід встановити прапорець Use firewall. При використанні захищеної версії протоколу SFTP встановіть прапорець Use Secure FTP (SFTP).

Все, що ми розглядали дотепер, це початкові налаштування сайту, після яких можна приступати безпосередньо до його побудови. Натискаємо кнопку ОК. Dreamweaver сам створить папку сайту, а ім'я сайту з'явиться в списку у вікні Manage Sites. Тепер папку сайту і все що зберігаються в ній документи можна буде побачити на панелі File.

Dreamweaver - програма з багатодокументним інтерфейсом або просто багатодокументна програма. Це означає, що ви можете відкрити в одному і тому ж вікні програми відразу кілька документів. В цьому випадку вікна, що містять відкриті документи, відкриваються всередині великого вікна самої програми. До багатодокументним додатків також відносяться Microsoft Word і Adobe Photoshop.

На відміну від них, програми з однодокументним інтерфейсом (однодокументний програми) можуть відкрити тільки один документ; щоб відкрити другий, потрібно запустити другу копію програми. Прикладами однодокументних застосувань є, зокрема, текстовий редактор Microsoft WordPad і графічний редактор Microsoft Paint, що поставляються в складі Windows.

Головне вікно служить "вмістилищем" для превеликого безлічі інших вікон, що містять як відкриті Web-сторінки, так і різні інструменти, призначені для роботи з ними. Також в головному вікні знаходиться рядок головного меню, за допомогою якого ви зможете отримати доступ до всіх можливостей Dreamweaver.

Вікно документа Dreamweaver служить для відображення відкритої Web-сторінки Як і будь-яка інша багатодокументна програма, Dreamweaver може відкрити в головному вікні скільки завгодно вікон документів. Ви можете переміщати, згортати і розгортати ці вікна і змінювати їх розміри, в загальному, проробляти з ним ті ж маніпуляції, що і з будь-яким іншим вікном Windows. Єдиний виняток: ви не можете "витягнути" жодне з цих

вікон за межі головного вікна програми (його ще називають батьківським вікном).

Питання для закріплення.

1. Що таке Dreamweaver?
2. В якому вікні налаштовуються властивості веб-сайту?
3. Назвіть основні налаштування сайту.

Рекомендована література:

- McFarland, David Sawyer. Dreamweaver CS6: The Missing Manual. "O'Reilly Media, Inc.", 2012.
- Aryal, S. (2019). Bootstrap: a front-end framework for responsive web design.
- Lopuck, Lisa. Web design for dummies. John Wiley & Sons, 2012.

Reference

1. Zachko O.B. "Development of a simulation model of safety management in the projects for creating sites with mass gathering of people." / O.B. Zachko, R.R. Golovaty, A.V. Yevdokymova. Eastern-European Journal of Enterprise Technologies, 2 (3) (2017): 15-24.
2. Мультиагентна модель управління безпекою при плануванні проектів створення об'єктів з масовим перебуванням людей / О. Б. Зачко Р. Р. Головатий // Вісник НТУ «ХП». Серія: Стратегічне управління, управління портфелями, програмами та проектами. – Х. : НТУ «ХП», 2017. – № 2 (1224). – С. 46–51. – Бібліогр.: 23 назв. – ISSN 2311-4738.
3. Рак Ю. П. Формалізація предметної області визначення «Об'єкт з масовим перебуванням людей» при реалізації безпеко-орієнтованих проектів / Ю. П. Рак, Р. Р. Головатий, Д. С. Кобилкін // Вісник Львівського державного університету безпеки життєдіяльності. – 2015. № 12. – С. 217 – 227.
4. Рак Ю.П. Класифікація та комплексна цінність стану торгово-розважальних центрів: проектно-орієнтований підхід / Ю.П. Рак, Р.Р. Головатий // Вісник НТУ «ХП» Серія: Стратегічне управління, управління портфелями, програмами та проектами. – Х.: НТУ «ХП», 2016. - №2 (1174). – С. 31-35.
5. Golovaty R. R. Safety management in project of creation the shopping malls // R. Golovaty // News of Science and Education: Sheffield. – 2016 –№ 20 (44) – P. 75–79.
6. Рак Ю. П. Безпеко-орієнтоване управління регіональними проектами захисту критичних інфраструктур засобами Системи 112 / Ю. П. Рак, О. Б. Зачко, Д. С. Кобилкін, Р. Р. Головатий // Управління проектами та розвиток виробництва: Зб. наук. пр. – Луганськ: вид-во СНУ ім. В. Даля. – 2016. № 1 (57). – С. 49 – 55.

7. Рак Ю.П. Формування проектів методом візуалізації інформації для підвищення стану безпеки торгово-розважальних центрів / Ю.П. Рак, Р.Р. Головатий // Управління проектами у розвитку суспільства: зб. тез доповідей XII Міжнар. конф. – Київ: КНУБА, 2015. – С. 226 – 228.
8. Рак Ю.П. Управління часом в проектах підвищення безпеки експлуатації в торгово-розважальних центрів / Ю.П. Рак, Р.Р. Головатий // Управління розвитком технологій: зб. тез доповідей II Міжнар. наук.-практ. конф. – Київ: КНУБА, 2015. – С. 74 – 76.
9. Рак Ю.П. Управління семантичним ядром оцінки визначень "ОБ'ЄКТ З МАСОВИМ ПЕРЕБУВАННЯМ ЛЮДЕЙ" на основі методу частотно-рангового розподілу / Ю.П. Рак, Р.Р. Головатий // Управління проектами: стан та перспективи: матер. XI міжнар. наук.-практ. конф. – Миколаїв: МНУК, 2015 - С. 186 – 187.
10. Рак Ю.П. Використання методу BSMNI в проектах безпечної експлуатації об'єктів з масовим перебуванням людей / Ю.П. Рак, Р.Р. Головатий // Управління проектами: інновації, нелінійність, синергетика: матер. VI міжнар. наук.-практ. конф. – Одеса: ОДАБА, 2015 - С. 130 – 132.
11. Рак Ю.П. Сервісна модель проектів створення об'єктів з масовим перебуванням людей / Ю.П. Рак, Р.Р. Головатий // Управління проектами у розвитку суспільства: зб. тез доповідей XIII Міжнар. конф. – Київ: КНУБА, 2016. – С. 207 – 208.
12. Головатий Р.Р. Агентне моделювання в проектах створення об'єктів з масовим перебуванням людей / Р.Р. Головатий // Управління проектами, програмами, портфелями: Тези доповідей I Міжнародної науково-практичної конференції : [у 2т.]. // Том. 2. – Одеса: ОНПУ., 2016 – С. 25-27.
13. Головатий Р.Р. Управління зацікавленими сторонами проекту безпечної експлуатації торгово-розважальних центрів // Р.Р. Головатий // III

Міжнародна науково-практична конференції «Інформаційні технології та взаємодії» (IT & I) // Київ: НУ ім. Т.Г. Шевченка, 2016 – С.55 – 57.

14. Зачко О.Б., Головатий Р.Р. Імітаційне моделювання потоку відвідувачів торгово-розважального центру / О.Б. Зачко, Р.Р. Головатий // Управління проектами: стан та перспективи: матер. XII міжнар. наук.-прак. конф. – Миколаїв: МНУК, 2016 - С. 96 – 98.
15. Зачко О.Б., Головатий Р.Р. Імітаційне моделювання пішоходних потоків в проектах створення об'єктів з масовим перебуванням людей / О.Б. Зачко, Р.Р. Головатий // I Всеукраїнська науково-практична конференція «Проблеми застосування інформаційних технологій, спеціальних технічних засобів у діяльності ОВС і навчальному процесі». – Львів: ЛДУВС, 2016 – С. 77-79.
16. Зачко О.Б., Кобилкін Д.С., Головатий Р.Р. Structural model of projects management of safety providing at objects with mass stay of people / О.Б. Зачко, Д.С. Кобилкін, Р.Р. Головатий // XII Міжнародна науково-практична конференція молодих вчених, курсантів та студентів. – Львів: ЛДУ БЖД, 2016 – С. 49-51.
17. Зачко О.Б., Головатий Р.Р. Інновінг управління проектами створення об'єктів з масовим перебуванням людей засобами безпеко-орієнтованого підходу // О.Б. Зачко, Р.Р. Головатий // XIV Міжнародна науково-практична конференція «Управління проектами у розвитку суспільства». – Київ, 2017 – С.121-123.