

# **ОСНОВИ ІНТЕРНЕТ**

Методичні рекомендації до вивчення дисципліни "Основи Інтернет"  
для студентів спеціальності "Інформатика"

**Ю.С. Павленко, Т.О. Гришанович**

## **ОСНОВИ ІНТЕРНЕТ**

Методичні рекомендації до вивчення дисципліни "Основи Інтернет"  
для студентів спеціальності "Інформатика"

УДК 004.738.5 (072)  
ББК 32.973.202 я 73-9  
П 12

*Рекомендовано до друку вченою радою  
Волинського національного університету імені Лесі Українки  
(протокол № 9 від 23.05.2012 р.)*

**Рецензенти:**

**Волошина Т.В.** – к.ф.-м.н., доц. кафедри алгебри та геометрії Волинського національного університету імені Лесі Українки

**Федонюк А.А.** – к.ф.-м.н., доц. кафедри вищої математики та інформатики Волинського національного університету імені Лесі Українки

Сучасний стан розвитку інформаційних технологій в різних галузях суспільного життя вимагає знання та вміння користуватися одним з найбільш популярних засобів доступу до інформаційних, комунікаційних та розважальних ресурсів – Інтернетом.

Дані вказівки призначені для студентів другого курсу спеціальності "Інформатика" для самостійного вивчення курсу "Основи Інтернет": оволодіння теоретичним матеріалом та практичними навичками роботи в мережі Інтернет. Цей матеріал також може бути використаний студентами інших спеціальностей, які вивчають роботу в Інтернет та створення веб-сторінок засобами HTML.

Посібник містить теоретичні відомості про структуру та сервіси Інтернет, мову розмітки гіпертекстових документів HTML, технологію каскадних таблиць стилів CSS. Наведений матеріал є достатнім для виконання завдань лабораторних робіт. В кінці сформульовані питання та завдання для самостійної роботи студентів, які дозволять детальніше ознайомитись із деякими сервісами мережі, програмними продуктами для роботи в мережі та розробки веб-сторінок.

УДК 004.738.5 (072)  
ББК 32.973.202 я 73-9  
П 12

© Павленко Ю.С., Гришанович Т.О., 2012  
© Волинський національний університет  
імені Лесі Українки

## Зміст

Тема 1. Поняття про мережу Інтернет. Історичні передумови та розвиток Інтернет .....	6
Тема 2. Комп'ютерні мережі. Локальні комп'ютерні мережі. Глобальні комп'ютерні мережі. Способи підключення до мережі Інтернет .....	9
Тема 3. Модель відкритих систем OSI .....	11
Тема 4. Стек протоколів TCP/IP. Адресація в мережі Інтернет .....	18
Тема 5. Сервіси мережі Інтернет. Сервіс WWW .....	23
Тема 6. Пошук в Інтернет. Складові та функціонування пошукових систем .....	29
Тема 7. Сервіси мережі Інтернет: електронна пошта, .....	35
сервіс FTP, файлообмінні мережі .....	35
Тема 8. Сервіси Інтернет: соціальні мережі, служба миттєвих повідомлень, сервіси Google, RSS .....	44
Тема 9. Мова гіпертекстової розмітки HTML. Структура HTML-документу. Робота з текстом в HTML .....	50
Тема 10. Мова гіпертекстової розмітки HTML. Створення таблиць. Робота з графічними об'єктами. Створення гіперпосилань .....	55
Тема 11. Мова гіпертекстової розмітки HTML. Фрейми. Карта сайту. Форми....	60
Тема 12. Каскадні таблиці стилів CSS. Поняття XHTML .....	67
Лабораторна робота № 1 .....	74
Лабораторна робота № 2 .....	75
Лабораторна робота № 3 .....	76
Лабораторна робота № 4 .....	77
Лабораторна робота № 5 .....	78
Лабораторна робота № 6 .....	79
Лабораторна робота № 7 .....	81
Лабораторна робота № 8 .....	82
Лабораторна робота № 9 .....	84
Лабораторна робота № 10 .....	84
Лабораторна робота № 11 .....	85

Лабораторна робота № 12.....	86
Лабораторна робота № 13.....	87
Лабораторна робота № 14.....	89
Запитання та завдання для самостійної роботи .....	91
Завдання для індивідуальної роботи .....	93
Список рекомендованих джерел.....	94
Додаток А.....	96

## Тема 1

### Поняття про мережу Інтернет. Історичні передумови та розвиток Інтернет

Існують різні підходи до визначення поняття "Інтернет".

24 жовтня 1995 року Федеральна Рада Мереж (ФРМ) одноголосно прийняла резолюцію, яка визначає термін Інтернет. Це визначення було створено в результаті консультацій з лідерами спільнот фахівців Інтернет та фахівців з питань Прав Інтелектуальної Власності (ПІВ).

#### РЕЗОЛЮЦІЯ:

“Федеральна Рада Мереж (ФРМ) затверджує нижченаведений вислів, як такий, що виражає визначення терміну “Інтернет”.

“Інтернет” визначає глобальну інформаційну систему, яка –

(i) є логічно взаємозв’язаною єдиним адресним простором, який базується на Інтернет–протоколі (IP) або його майбутніх розширеннях/наступних версіях;

(ii) спроможна підтримувати комунікації шляхом використання сімейства протоколів: Протоколу Управління Передачею/Інтернет-протоколу (TCP/IP) або його (сімейства) майбутніх розширень/наступних версій, та/або інших IP-сумісних протоколів; та

(iii) надає, використовує або робить доступними, публічно або персонально, високорівневі сервіси, які накладені на описані вище комунікації та супутню їм інфраструктуру.

В абзаці п’ятнадцятому статті 1 Закону України "Про телекомунікації" зазначено: Інтернет – всесвітня інформаційна система загального доступу, яка логічно зв’язана глобальним адресним простором та базується на Інтернет-протоколі, визначеному міжнародними стандартами (<http://zakon2.rada.gov.ua/laws/show/1280-15>).

Перелік ключових дат.

**1969 року** Міністерство оборони США започаткувало розробку проекту, який мав метою створення надійної системи передачі інформації на випадок

війни. Агенство передових досліджень США (ARPA) запропонувало розробити для цього комп'ютерну мережу. Її розробка була доручена:

- Каліфорнійському університетові Лос-Анжелеса;
- Стенфордському дослідному центру;
- Університету штату Юта;
- Університету штату Каліфорнія в Санта-Барбарі.

Ця мережа була названа **ARPANET** (Advanced Research Projects Agency Network – Мережа Агенства передових досліджень). В рамках проекту мережа об'єднала вищеназвані заклади; всі роботи по її створенню фінансувались за рахунок Міністерства оборони США. Потім мережа ARPANET почала активно рости й розвиватись; її дедалі ширше почали використовувати вчені із різних областей науки.

**1 вересня 1969 року** – встановлено перший сервер ARPANET у Каліфорнійському університеті в Лос-Анжелесі. Комп'ютер "Honeywell 516" мав 12 кілобайт оперативної пам'яті.

До **1971 року** була розроблена перша програма для відправки електронної пошти по мережі.

**1973 р.** – до мережі через трансатлантичний кабель під'єднано перші іноземні організації з Великобританії та Норвегії; мережа стала міжнародною.

**1970-і роки** – мережа в основному використовувалась для пересилки електронної пошти, тоді ж з'явилися перші списки поштових розсилок, групи новин та дошки оголошень. Але в ті часи мережа ще не могла легко взаємодіяти з іншими мережами, які були побудовані на інших технічних стандартах. До кінця 1970-х років почали активно розвиватись протоколи передачі даних, що були стандартизовані у 1982-1983 роках.

**1 січня 1983 року** – мережа ARPANET перейшла з протоколу NCP на **протокол TCP/IP**. Саме у 1983 році за мережею ARPANET закріпився термін "Інтернет".

**1984 р.** – розроблена **система доменних імен** (Domain Name System, **DNS**).

**1989 р.** – в Європі, в Європейському центрі ядерних досліджень (Conseil Européen pour la Recherche Nucléaire, CERN) була сформульована концепція WWW. Її запропонував британський вчений Тім Бернерс-Лі, він протягом двох років розробляв протокол HTTP, мову гіпертекстової розмітки HTML та ідентифікатори URI.

**1990 р.** – мережа ARPANET припинила своє існування, програвши конкуренцію NSFNet. Тоді ж було зафіксовано перше під'єднання до Інтернету по телефонній лінії ("додзвонювання", Dial-up access).

**1991 р.** – WWW став доступним в Інтернет, а в 1993 році з'явився оглядач WWW (web-browser) **NCSA Mosaic**.

**1995 р.** – NSFNet повернулась до ролі дослідницької мережі; маршрутизацією всього трафіку Інтернету тепер займались мережеві провайдери (постачальники послуг), а не суперкомп'ютери Національного наукового фонду.

В цьому ж році WWW став основним постачальником інформації в Інтернет, обігнавши по об'єму трафіку протокол передачі файлів FTP; було сформовано Консорціум всесвітньої павутини (WWW Consortium, **W3C**). Можна сказати, що сервіс WWW перетворив Інтернет і створив його сучасний вигляд. З **1996 року** сервіс WWW майже повністю замінив собою поняття "Інтернет".

У **1998 році** папа римський Іоанн Павло II заснував **Всесвітній день Інтернету – 30 вересня**.

Перші користувачі мережі Інтернет з'явилися на території України ще за радянських часів.

**1990 року** було створено перші три вузли доступу до мережі. Тоді ж було розпочато деяку підтримку української частини доменного простору мережі Інтернет – домену .ua. Спочатку домен підтримувався неформально при маршрутизації пошти співробітниками московського АТ "Релком", на той час відомого як Інтернет-вузол Інституту атомної енергії ім. Курчатова.

**1 грудня 1992 року** – домен .ua було офіційно делеговано Україні. З цього часу прийнято відраховувати історію українського сегменту мережі Інтернет.



Адміністрування домену доручено співробітникам Київського ТОВ "Комунікаційні Системи".

### *Запитання для самоконтролю*

1. *Що таке Інтернет?*
2. *В якому році було здійснено перехід на протокол TCP/IP?*
3. *Який день вважається Всесвітнім днем Інтернету?*
4. *В якому році розроблена система доменних імен?*

## **Тема 2**

### **Комп'ютерні мережі. Локальні комп'ютерні мережі. Глобальні комп'ютерні мережі. Способи підключення до мережі Інтернет**

**Комп'ютерна мережа** – це сукупність комп'ютерів, з'єднаних каналами передачі даних. Залежно від відстані між комп'ютерами розрізняють наступні обчислювальні мережі:

- локальні обчислювальні мережі (ЛОМ), Local Area Networks (LAN) – комп'ютерні мережі, розташовані в межах невеликої обмеженої території (будівлі або в сусідніх будівлях) не більше 10-15 км;
- територіальні обчислювальні мережі, які охоплюють значний географічний простір. До територіальних мереж можна віднести мережі: регіональні (MAN – Metropolitan Area Network) і глобальні (WAN – Wide Area Network), що мають, відповідно регіональні або глобальні масштаби. Регіональні мережі зв'язують абонентів району, міста або області. Глобальні мережі об'єднують абонентів, що знаходяться в різних країнах або континентах;
- корпоративні обчислювальні мережі – це сукупність зв'язаних між собою локальних обчислювальних мереж, що охоплюють територію, на якій розміщені будівлі підприємства.

За призначенням комп'ютерні мережі поділяються на:

- обчислювальні;

- інформаційні;
- змішані (інформаційно-обчислювальні).

За типом комп'ютерів, які входять до складу комп'ютерної мережі розрізняють:

- однорідні комп'ютерні мережі, які складаються із програмно-однакових ЕОМ;

- неоднорідні, до складу яких входять програмно-несумісні комп'ютери.

Локальні обчислювальні мережі забезпечують:

- розподіл даних (Data Sharing). Дані в ЛОМ зберігаються на центральному ПК і можуть бути доступними на робочих станціях, тому на кожному робочому місці не треба мати накопичувачів для зберігання однієї і тієї ж інформації;

- розподіл інформаційних і технічних ресурсів (Resource Sharing): логічні диски та інші зовнішні запам'ятовуючі пристрої; каталоги і файли, що в них містяться; підключені до ПК пристрої (дозволяє економно використовувати ресурси);

- розподіл програм (Software Sharing). Всі користувачі ЛОМ можуть мати спільний доступ до програм (мережним версіям);

- обмін повідомленнями.

Локальні обчислювальні мережі можуть відрізнятися архітектурою (Ethernet, Token Ring, FDDI і т.д.) і топологією (шинна, кільцева, "зірка"). Вибір типу ЛОМ залежить від потреб користувачів і фінансових можливостей підприємства.

Мережева технологія (архітектура) – це злагоджений набір стандартних протоколів і програмно-апаратних засобів, які їх реалізують, достатній для побудови локальної обчислювальної мережі. Мережні технології називають базовими технологіями або мережною архітектурою локальних мереж.

Мережева технологія або архітектура визначає топологію і метод доступу до середовища передачі даних, кабельну систему або середовище передачі

даних, формат мережних кадрів, тип кодування сигналів, швидкість передачі в локальній мережі.

Глобальні мережі призначені, як і ЛОМ, для надання послуг, але значно більшій кількості користувачів, що знаходяться на великій території.

Компанії, що здійснюють підтримку функціонування мережі, називаються операторами мережі, а компанії, які надають платні послуги абонентам мережі, називаються провайдерами або постачальниками послуг.

### ***Запитання для самоконтролю***

- 1. Що таке комп'ютерна мережа?*
- 2. Як класифікують комп'ютерні мережі?*
- 3. Які можливості забезпечують комп'ютерні мережі?*
- 4. Що таке мережна технологія?*
- 5. Що таке топологія мережі?*
- 6. Які бувають мережні технології?*
- 7. Перерахувати топології мереж.*

## **Тема 3**

### **Модель відкритих систем OSI**

Комп'ютерні мережі, як правило, складаються з різного устаткування різних виробників, і без прийняття всіма виробниками загальноприйнятих правил побудови персональних комп'ютерів і мережевого устаткування, забезпечити нормальне функціонування мереж було б неможливим. Тобто необхідний єдиний уніфікований стандарт, який визначав би алгоритм передачі інформації в мережах. В сучасних обчислювальних мережах роль такого стандарту виконують мережні **протоколи**.

Оскільки описати єдиним протоколом взаємодії між пристроями в мережі неможливо, то необхідно розділити процес мережевої взаємодії на ряд концептуальних рівнів (модулів), визначити функції для кожного модуля і

порядок їх взаємодії, застосувавши метод декомпозиції. **Декомпозиція** допускає чітке визначення функції кожного рівня і інтерфейсів між рівнями.

Обидва учасники мережного обміну повинні узгодити багато домовленостей, які повинні бути прийняті для всіх рівнів, починаючи від найнижчого – рівня передачі бітів, до найвищого, який реалізує сервіс для користувача.

Взаємодія однойменних функціональних рівнів по горизонталі здійснюється за допомогою протоколів. **Протоколом** називається набір правил і методів взаємодії однойменних функціональних рівнів об'єктів мережного обміну.

Взаємодія функціональних рівнів по вертикалі здійснюється через інтерфейси. **Інтерфейс** визначає набір функцій, які нижчий рівень надає вищому рівню.

Механізм передачі деякого пакету інформації через мережу від клієнтської програми, що працює на одному комп'ютері ПК1, до клієнтської програми, що працює на іншому комп'ютері ПК2, можна умовно представити у вигляді послідовної пересилки цього пакету зверху вниз від верхнього рівня, що забезпечує взаємодію з призначеним для користувача додатком, до нижнього рівня, який організовує інтерфейс з мережею, його трансляції на комп'ютер ПК 2 і зворотної передачі верхньому рівню вже на комп'ютер ПК 2 (рис. 1).

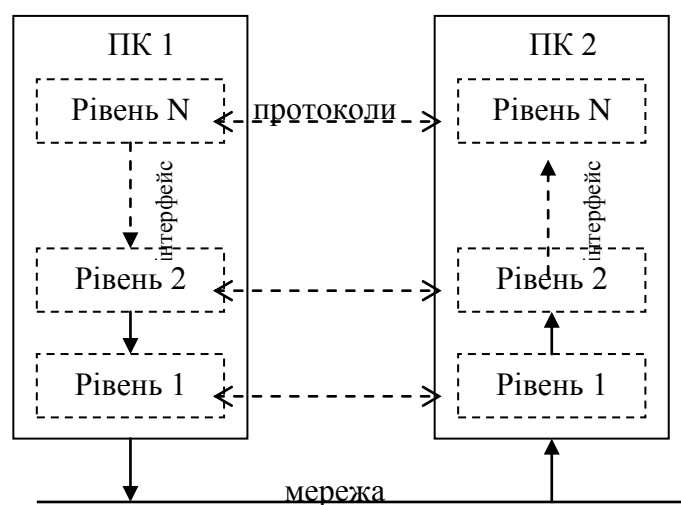


Рисунок 1 - Механізм передачі інформації від одного комп'ютера до іншого

Комунікаційні протоколи можуть бути реалізовані як **програмно**, так і **апаратно**. Протоколи нижніх рівнів часто реалізуються комбінацією програмних і апаратних засобів, а протоколи верхніх рівнів – як правило, тільки програмними засобами.

Протоколи реалізуються не тільки комп'ютерами, а й іншими мережевими пристроями – концентраторами, мостами, комутаторами, маршрутизаторами та ін. Залежно від типу пристрою в ньому повинні бути вбудовані засоби, що реалізують той або інший набір протоколів.

Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів в мережі, називається **стеком комунікаційних протоколів**. В мережі Інтернет базовим набором протоколів є стек протоколів TCP/IP.

У **1978 році** Міжнародний комітет зі стандартизації (ISO) розробив стандарт архітектури ISO 7498, для об'єднання різних мереж. У розробці брало участь 7 комітетів, кожному з яких був відведений свій рівень.

В **1984 році** специфікація моделі OSI (Open System Interconnection) була переглянута й прийнята як міжнародний стандарт для мережових комунікацій. Еталонна модель OSI є визначальним документом концепції розробки відкритих стандартів для організації з'єднання систем. **Відкрита система** – це система, доступна для взаємодії з іншими системами відповідно до прийнятих стандартів.

OSI була розроблена з метою спрощення взаємодії пристроїв в мережах і:

- є рекомендаціями розробникам мереж і протоколів для побудови стандартів сумісних мережних програмних продуктів і служить базою для виробників при розробці сумісного мережного устаткування;
- визначає сім рівнів взаємодії систем в мережах з комунікацією пакетів, дає їм стандартні імена і вказує, які функції повинен виконувати кожний рівень. Кожний рівень функціонує незалежно від вищих і нижчих рівнів. Кожний рівень може спілкуватися з безпосереднім сусіднім рівнем, проте він повністю ізольований від прямого звернення до наступних рівнів;

- описує тільки системні засоби взаємодії, які реалізуються операційною системою, системними утилітами, системними апаратними засобами. Модель не включає засобу взаємодії додатків кінцевих користувачів.

При розробці моделі виділення рівнів базувалося на наступних принципах:

- кожен рівень повинен виконувати окрему функцію;
- потік інформації між рівнями повинен бути мінімізований;
- функції рівнів повинні бути зручні для визначення міжнародних стандартів;
- кількість рівнів повинна бути достатньою для поділу функцій, але не надлишковою.

Виділяють наступні рівні OSI:

- фізичний;
- канальний;
- мережевий;
- транспортний;
- сеансовий;
- представлення;
- прикладний.

**7. Прикладний рівень (Application layer)** забезпечує взаємодію мережі й користувача.

На цьому рівні реалізуються передача файлів, віддалений термінальний доступ, електронна передача повідомлень, керування мережею та ін. Одиниця даних, якою оперує рівень називається повідомленням. Також даний рівень відповідає за передачу службової інформації, надає додаткам інформацію про помилки й формує запити до **рівня представлення**.

**6. Рівень представлення (Presentation layer)** має справу із синтаксисом і семантикою переданої інформації, тобто тут встановлюється взаєморозуміння двох сполучених комп'ютерів відносно того, як вони представляють і розуміють після одержання передану інформацію. При необхідності виконується

перетворення форматів даних в деякий загальний формат подання, а на боці отримувача, відповідно, виконується зворотне перетворення.

Запити додатків, отримані з рівня додатків, він перетворює у формат для передачі по мережі, а отримані з мережі дані перетворює у формат, зрозумілий додаткам. На цьому рівні може здійснюватися стиснення/розпакування або кодування/розкодування даних, підтримка мережеских файлових систем (NFS), абстрактних структур даних, а також перенаправлення запитів іншому мережному ресурсу, якщо вони не можуть бути оброблені локально.

**5. Сеансовий рівень (Session layer)** координує взаємодію користувачів, що встановлюють зв'язок, відповідає за підтримку сеансу зв'язку, дозволяючи додаткам взаємодіяти між собою тривалий час. Встановлює зв'язок, оперує ним, керує створенням/ завершенням сеансу, відновлює аварійно закінчені сеанси, управляє обміном інформацією, синхронізацією завдань, визначенням права на передачу даних і підтримкою сеансу в періоди неактивності додатків. Синхронізація передачі забезпечується вставкою у потік даних контрольних точок, починаючи з яких відновлюється процес при порушенні взаємодії.

Цей рівень відповідальний за картографію мережі – він перетворює регіональні (доменні) комп'ютерні імена в числові адреси, і навпаки.

**4. Транспортний рівень (Transport layer)** призначений для доставки даних без помилок, втрат і дублювання в тій послідовності, як вони були передані. При цьому неважливо, які дані передаються, звідки й куди, тобто він надає сам механізм передачі. Блоки даних він розділяє на фрагменти, розмір яких залежить від протоколу, короткі поєднує в один, довгі розбиває.

Регламентує пересилання пакетів повідомлень між процесами, що виконуються на комп'ютерах мережі. Починає і завершує організацію передачі даних: контролює на наскрізній основі потік даних, що проходить по маршруті, визначеному п'ятим рівнем: правильність передачі блоків даних, правильність доставки в потрібний пункт призначення, їх комплектність, цілісність, порядок слідування. Збирає інформацію з блоків у її попередній вигляд.

Перевіряє правильність доставки й адресації, повторює посилку дейтаграми, якщо не надійшов відгук. Іншими словами, задачею даного рівня є правильно і точно передати інформацію з будь-якої точки в довільну точку мережі.

**3. Мережевий рівень (Network layer)** призначений для визначення шляху передачі даних. Відповідає за трансляцію логічних адрес й імен у фізичні, визначення найкоротших маршрутів, комутацію й маршрутизацію пакетів, відстеження неполадок і заторів у мережі. Введений для того, щоб зберегти простоту передачі даних для типових топологій з одного боку, а з іншого – дозволити використання довільних топологій. На цьому рівні працює такий мережний пристрій, як маршрутизатор.

Мережі об'єднуються між собою спеціальним пристроєм, який називається **маршрутизатором**. Він збирає інформацію про топологію міжмережових з'єднань і на її основі пересилає пакети мережевого рівня в пункт призначення. Для того, щоб передати повідомлення від відправника, який знаходиться в одній мережі, отримувачу з іншої, потрібно здійснити кілька транзитних передач (hops) між мережами, кожен раз вибираючи маршрут, який найбільше підходить в даному випадку. Отже, маршрут – це послідовність маршрутизаторів, через які проходить пакет.

Проблема відшукування найкращого шляху називається маршрутизацією і її вирішення є головним завданням мережевого рівня. Проблема ускладнюється, оскільки найкоротший шлях не завжди є найкращим. Часто критерієм вибору є час передачі даних по маршруту, що залежить від пропускнуєї спроможності каналів зв'язку і інтенсивності трафіку, яка може змінюватися в часі; а також надійність передачі.

**2. Канальний рівень (Data Link layer)** перевіряє доступність середовища передачі, реалізує механізми виявлення і корекції помилок, забезпечує коректність передачі кожного кадру (групи бітів):

- розміщає спеціальну послідовність біт в початок і кінець кожного кадру, щоб відмітити його;



- обчислює контрольну суму, сумуючи всі байти кадру визначеним способом і додає цю контрольну суму до кадру;
- після надходження кадру отримувач знову обчислює контрольну суму отриманих даних і порівнює результат з контрольною сумою з кадру. Якщо вони співпадають, кадр вважається правильним і приймається. Інакше – фіксується помилка.

В локальних мережах протоколи канального рівня використовуються мостами, комутаторами і маршрутизаторами. В комп'ютерах функції канального рівня реалізуються мережевими адаптерами та їх драйверами.

На цьому рівні цифрові дані з пакетів, створених раніше, накладаються на фізичні сигнали, що генеруються мережевим пристроєм, який використовується, і змінюють їх (модулюють). Ця операція відбувається під керуванням програм, встановлених разом з драйвером.

**1. Фізичний рівень (Physical layer)** включає фізичні аспекти передачі двійкової інформації по лінії зв'язку. Детально описує, наприклад, напруги, частоти, природу середовища передачі.

Призначений безпосередньо для передачі потоку даних. Здійснює передачу електричних або оптичних сигналів у кабель і відповідно їхній прийом і перетворення в біти даних відповідно до методів кодування цифрових сигналів. Інакше кажучи, здійснює інтерфейс між мережним носієм і мережним пристроєм. На цьому рівні працюють концентратори й повторювачі (ретранслятори) сигналу.

Функції фізичного рівня реалізуються у всіх пристроях, які підключені до мережі. З боку комп'ютера вони виконуються мережевим адаптером.

Таблиця 1. Розподіл протоколів по рівнях моделі OSI

Рівень OSI	Протоколи
прикладний	HTTP, gopher, Telnet, DNS, SMTP, SNMP, CMIP, FTP, TFTP, SSH, IRC, AIM, NFS, NNTP, NTP, SNTP, XMPP, FTAM, APPC, X.400, X.500, AFP, LDAP, SIP, ITMS, Modbus TCP, BACnet IP, IMAP, POP3, SMB, MFTP, BitTorrent, e2k, PROFIBUS і т.д.

представлення	HTTP/HTML, ASN.1, XML, TDI, XDR, SNMP, FTP, Telnet, SMTP, NCP, AFP
сеансовий	ASP, ADSP, DLC, Named Pipes, NBT, NetBIOS, NWLink, Printer Access Protocol, Zone Information Protocol, SSL, TLS, SOCKS
транспортний	TCP, UDP, NetBEUI, AEP, ATP, IL, NBP, RTMP, SMB, SPX, SCTP, DCCP, RTP, STP, TFTP
мережний	IP, IPv6, ICMP, IGMP, IPX, NWLink, NetBEUI, DDP, IPSec, ARP, RARP, DHCP, Boot, SKIP
каналний	ARCnet, ATM, DTM, SLIP, SMDS, Ethernet, FDDI, Frame Relay, LocalTalk, Token ring, PPP, PPPo, StarLan, WiFi, PPTP, L2F, L2TP, PROFIBUS
фізичний	RS-232, RS-422, RS-423, RS-449, RS-485, ITU-T, xDSL, ISDN та ін.

### *Запитання для самоконтролю*

1. *Що таке протокол?*
2. *Що таке інтерфейс?*
3. *Як можуть бути реалізовані комунікаційні протоколи?*
4. *Що таке модель відкритих систем OSI?*
5. *Що таке відкрита система?*
6. *Назвати рівні моделі OSI.*
7. *Охарактеризувати кожен із рівнів OSI.*

## **Тема 4**

### **Стек протоколів TCP/IP. Адресація в мережі Інтернет**

**Протокол** – це набір правил і процедур, що регулюють порядок здійснення зв'язку. Протоколи задають засоби передачі повідомлень і

опрацювання помилок у мережі, а також дозволяють розробляти стандарти, не прив'язані до конкретної апаратної платформи. Всі параметри – від швидкості передачі даних до методів адресації при транспортуванні окремих повідомлень – задаються протоколами, що використовуються в даній конкретній мережі.

Існує кілька стандартних наборів (або, як їх ще називають, стеків) протоколів:

- набір протоколів ISO/OSI;
- IBM System Network Architecture (SNA);
- Digital DECnet;
- Novell NetWare;
- Apple AppleTalk;
- набір протоколів глобальної мережі Internet, TCP/IP.

TCP/IP – Transmission Control Protocol/Internet Protocol (Протокол керування передачею/Протокол Internet) – фактично не один протокол, а декілька, серед яких TCP і IP – два основних. TCP/IP становлять базовий набір протоколів Інтернету, відповідальний за розбивку вихідного повідомлення на пакети (TCP), доставку пакетів на вузол адресата (IP) і збирання (відновлення) вихідного повідомлення з пакетів (TCP).

Інформація, що надсилається по IP-мережах, розбивається на порції, що називаються пакетами (технічні, апаратні обмеження). В одному пакеті звичайно надсилаються від 1 до 1500 символів інформації.

Однак, при користуванні IP-протоколом виникають відразу декілька проблем:

- в більшості випадків, обсяг інформації, що передається, перевищує 1500 символів;
- при передачі інформації може виникнути помилка;
- послідовність доставки пакетів може бути порушена.

Протокол TCP поділяє інформацію на порції – пакети, кожному з яких надає номер для правильного відновлення інформації під час одержання (рис. 2). Протокол IP додає до кожного пакета службову інформацію з адресами

відправника і одержувача(ів), забезпечуючи доставку всіх пакетів одержувачеві. Окремі пакети можуть подорожувати різними шляхами Інтернету та дістатися до одержувача у будь-якому порядку. По надходженні всіх пакетів протокол TCP розміщує їх один за одним і забезпечує складання повідомлення. Якщо деякі пакети загубилися, протокол TCP вирішує і цю проблему. Маршрутами руху пакетів мережею керує маршрутизатор.

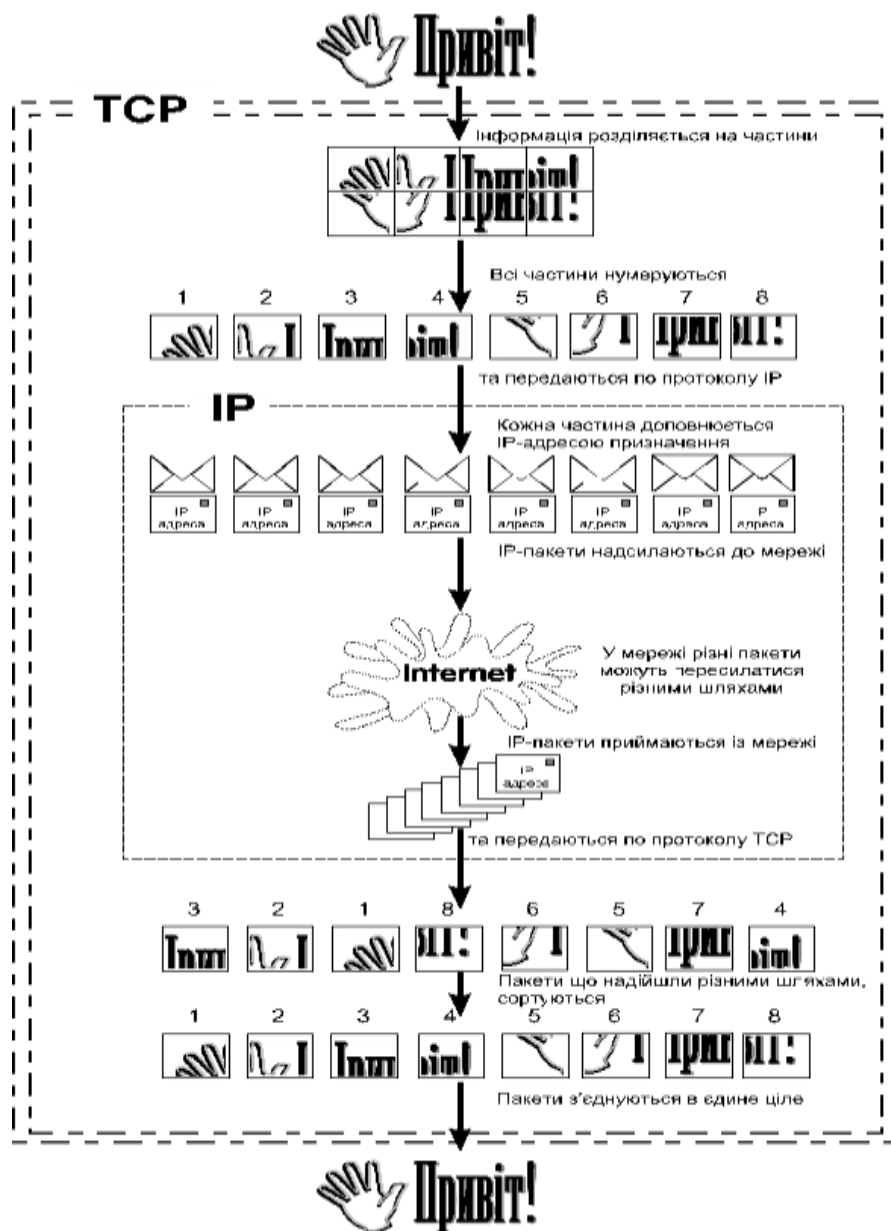


Рисунок 2 - Схема роботи протоколу TCP/IP

Багаторівнева структура стеку TCP/IP:

1. Прикладний рівень (прикладний + представлення + сеансовий).
2. Транспортний рівень.
3. Мережевий рівень.

#### 4. Рівень мережевих інтерфейсів (канальний + фізичний).

У стеку TCP/IP використовуються **три типи адрес**:

- локальні (апаратні),
- IP-адреси,
- символічні доменні імена.

**Локальна адреса** вузла визначається технологією, за допомогою якої побудована окрема мережа, в яку входить даний вузол. Для вузлів, що входять в локальні мережі – це MAC-адреса мережного адаптера або порту маршрутизатора, наприклад, 11-A0-17-3D-BC-01.

Ці адреси призначаються виробниками устаткування і є унікальними адресами, оскільки управляються централізовано. Для всіх існуючих технологій локальних мереж MAC-адреса має формат 6 байтів: старші 3 байти – ідентифікатор фірми виробника, а молодші 3 байти призначаються унікальним чином самим виробником.

**IP-адреси** – це основний тип адрес, на підставі яких мережевий рівень передає пакети між мережами. Ці адреси складаються з 4 байт, наприклад, 109.26.17.100. IP-адреса призначається адміністратором під час конфігурування комп'ютерів і маршрутизаторів.

**Символьні доменні імена** будуються по ієрархічній ознаці. Складові повного символічного імені в IP-мережах розділяються крапкою й перераховуються в наступному порядку: спочатку просте ім'я кінцевого вузла, далі – ім'я групи вузлів (наприклад, ім'я організації), далі – ім'я більшої групи (піддомена) і так до імені домена найвищого рівня.

Між доменним іменем і IP-адресою вузла немає ніякої алгоритмічної відповідності, тому необхідно використовувати додаткові таблиці або служби, щоб вузол мережі однозначно визначався як по доменному імені, так і по IP-адресі. У мережах TCP/IP використовується спеціальна розподілена служба Domain Name System (DNS), що встановлює цю відповідність на підставі створюваних адміністраторами мережі таблиць відповідності. Тому доменні імена називають також DNS-іменами.

IP-адреса має довжину 4 байти і записується у вигляді чотирьох чисел, що представляють значення кожного байта в десятковій формі й розділених крапками, наприклад, 128.10.2.30 – традиційна десяткова форма представлення адреси, а 10000000 00001010 00000010 00011110 – двійкова форма представлення цієї ж адреси.

Адреса складається із двох логічних частин – номера мережі й номери вузла в мережі. Значеннями перших біт адреси визначається, яка частина адреси відноситься до номера мережі, а яка – до номера вузла. Значення цих біт є також ознаками того, до якого класу відноситься та або інша IP-адреса.

Таблиця 2. Діапазон номерів мереж, що відповідають певному класу

Клас	Перші біти	Найменший номер мережі	Найбільший номер мережі	Кількість байт для номера мережі	Кількість байт для номера вузла	Кількість мереж	Кількість вузлів у мережі
A	0	1.0.0.0	126.0.0.0	1	3	$2^7$	$2^{24}$
B	10	128.0.0.0	191.255.0.0	2	2	$2^{14}$	$2^{16}$
C	110	192.0.1.0	223.255.255.0	3	1	$2^{21}$	$2^8$
D	1110	224.0.0.0	239.255.255.255			не визначено	
E	11110	240.0.0.0	247.255.255.255			не визначено	

Великі мережі одержують адреси класу А, середні – класу В, а малі – класу С.

**Особливі IP-адреси.** У протоколі IP існує кілька узгоджень про особливу інтерпретацію IP-адрес. Наступні типи адрес не використовуються для присвоєння комп'ютерам під час налаштування з'єднання з Інтернет:

- вся адреса складається тільки із двійкових нулів;
- в полі номера мережі записані тільки нулі;
- всі двійкові розряди IP-адреси рівні 1;
- в полі номера вузла призначення записані тільки одиниці.

При таких обмеженнях максимальна кількість вузлів, наведена в таблиці для мереж кожного класу, на практиці зменшена на 2.

Особливий смисл має IP-адреса, перший октет якої дорівнює 127. Вона використовується для тестування програм і взаємодії процесів у межах однієї машини. Коли програма посилає дані по IP-адресі 127.0.0.1, то утворюється так

звана "петля": дані не передаються по мережі, а повертаються модулям верхнього рівня як щойно прийняті. Тому в IP-мережі забороняється присвоювати машинам IP-адреси, що починаються з 127. Ця адреса має назву loopback.

Регулюванням питань, пов'язаних з доменними іменами, IP-адресами і іншими аспектами функціонування Інтернету займається Інтернет корпорація з присвоєння імен та номерів – Internet Corporation for Assigned Names and Numbers, ICANN – міжнародна некомерційна організація, створена восени 1998 за участю уряду США.

30 жовтня 2009 року ICANN схвалила процедуру реєстрації доменних імен, що використовують не тільки латиницю, але і символи національних алфавітів. Заявка на домен .укр подана в ICANN 16 листопада 2009.

### ***Запитання для самоконтролю***

- 1. Що таке TCP/IP?*
- 2. Які типи адрес використовуються в TCP/IP?*
- 3. Описати кожен із типів адрес.*
- 4. Перерахувати класи IP-адрес.*
- 5. Навести приклади "особливих" IP-адрес.*
- 6. Що таке DNS?*

## **Тема 5**

### **Сервіси мережі Інтернет. Сервіс WWW**

Служби (сервіси) Інтернет – це види послуг, що надаються серверами мережі Internet. В історії Інтернет існували різні види сервісів, одні з яких у даний час використовуються дуже активно, інші поступово втрачають свою популярність, а решта не використовуються на даний час взагалі.

Більшість сервісів Інтернет працюють із використанням принципу "клієнт-сервер". Це означає, що, коли користувач (клієнт) запускає програму, яка

запитує послугу (таку програму також називають "клієнтом"), вона з'єднується мережею з іншою програмою (таку програму називають "сервером"), запущеною на іншій машині мережі. Комп'ютер, що надає мережевi послуги, також називається сервером. Комп'ютер, що підключений до мережі і відповідає на запити, називається хостом (host).

Сервером може бути будь-який хост (навіть РС). Для цього на ньому повинні бути запущені відповідні програми, що надають сервісні мережевi послуги, або ці програми запускаються автоматично операційною системою. Хост, на якому працює користувач, називається локальним. Комп'ютер-хост повинен мати 32-розрядну ІР-адресу і ім'я в доменній формі.

Дати певну класифікацію сервісів, залежно від їхнього застосування, практично неможливо через унікальність кожного сервісу й одночасно невіддільність його від інших. Прийmemo таку класифікацію сервісів:

- інтерактивні (on-line);
- прямі;
- відкладеного читання (off-line).

Хоча цей поділ не є точним, оскільки, одним і тим самим сервісом (наприклад, FTP) можна користуватися як у режимі on-line, так і в режимі off-line через ftp-mail.

Сервіси відкладеного читання (off-line) найменш вимогливі до ресурсів комп'ютера і ліній зв'язку. Основною ознакою цих сервісів є та особливість, що запит і одержання інформації можуть бути сильно розділені в часі. До сервісу відкладеного читання можна віднести, наприклад, електронну пошту.

Прямі сервіси характерні тим, що інформація на запит клієнта повертається негайно. Однак від клієнта (одержувача інформації) не потрібно негайної реакції на отриману відповідь. До цього типу сервісів належить більшість служб мережі Інтернет (наприклад, WWW, FTP та ін.).

Сервіси інтерактивні характеризуються необхідністю негайної реакції на отриману інформацію, тобто одержувана інформація можна вважати запитом.

Як правило, виділяють наступні служби Інтернету:



- Веб (WWW):
  - Веб-форуми;
  - блоги;
  - вікі-проекти (в тому числі й Вікіпедія);
  - Інтернет-магазини;
  - Інтернет-аукціони;
- електронна пошта та списки розсилки;
- FTP-сервери;
- служба передавання миттєвих повідомлень;
- IRC;
- служба Telnet (віддалений доступ);
- IP-телефонія
- файлообмінні мережі та ін.

WWW – World Wide Web ("Всесвітня павутина") – розподілена система, що надає доступ до пов'язаних між собою документів, розташованих на різних комп'ютерах, підключених до Інтернету. WWW – сервіс прямого доступу, який потребує повноцінного підключення до Інтернет, і часто вимагає швидких ліній зв'язку (у випадку, коли документи, які користувач читає, містять багато графічної або іншої нетекстової інформації).

Проект WWW був започаткований в березні 1989 року в Європейській Лабораторії Фізики Елементарних Часток (European Laboratory for Particle Physics (CERN), Женева, Швейцарія). Тім Бернес-Лі запропонував новий спосіб обміну результатами досліджень і ідеями між організаціями. Такий обмін був дуже необхідний, оскільки члени цієї організації працювали в різних країнах. Пропонувалося використати просто систему гіпертексту для передачі документів і встановлення зв'язку між фізиками-ядерниками. Тоді ще не було планів застосування звуку або відео, і не розглядалася можливість передачі зображення. В результаті Бернерс-Лі написав дві програми: браузер, який був названий WorldWideWeb і перший веб-сервер, який працював на комп'ютері NEXTSTEP.

Перший веб-сервер в США був встановлений 12 грудня, 1991 р., в SLAC (Стенфордський центр лінійного прискорювача).

В січні 1995 було засновано World-Wide Web Consortium (W3C). W3C, що керується спільно з MIT/LCS у США, Keio University в Японії та INRIA у Франції, вже до 2002 мав більше, ніж 500 своїх організацій по всьому світу.

Ідея Т. Бернерс-Лі полягала в тому, щоб застосувати гіпертекстову модель до інформаційних ресурсів, які розподілені в мережі, і зробити це максимально простим способом. Вона базується на таких засадах:

- мова розмітки гіпертекстових документів HTML – HyperText Markup Language;
- універсальний спосіб адресації ресурсів в мережі URL – Universal Resource Locator;
- протокол обміну гіпертекстовою інформацією HTTP – HyperText Transfer Protocol.

Пізніше команда NCSA додала ще один компонент:

- універсальний інтерфейс шлюзів CGI – Common Gateway Interface.

Інформаційний WWW-сервер використовує гіпертекстову технологію. Для запису документів в гіпертексті використовується мова HTML, яка дозволяє управляти шрифтами, відступами, вставляти кольорові ілюстрації, підтримує виведення звуку і анімації. В стандарт мови також входить підтримка математичних формул.

URI визначає спосіб запису (кодування) адрес різних інформаційних ресурсів при зверненні до них із сторінок WWW. URI – це символічний рядок, який дозволяє ідентифікувати довільний ресурс: документ, зображення, файл, службу, скриньку електронної пошти та ін. Існує кілька схем ідентифікації всередині URI. Найвідомішими з них є URL і URN.

URL – Uniform Resource Locator – це URI, який не лише ідентифікує ресурс, але й надає інформацію про місцезнаходження ресурса.

URN – Uniform Resource Name – це URI, який ідентифікує ресурс в певному просторі імен.

URL використовується в гіпертекстових посиланнях і забезпечує доступ до розподілених ресурсів мережі. В URL можна адресувати як інші гіпертекстові документи формату HTML, так і ресурси e-mail, telnet, ftp та ін.

HTTP – мережевий протокол прикладного рівня передачі даних у вигляді текстових повідомлень. Основою протоколу HTTP є технологія клієнт-сервер. Протокол HTTP в даний час використовується в WWW для отримання інформації з веб-сайтів. Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URL в запиті клієнта. Звичайно такими ресурсами є файли, що зберігаються на сервері. Особливістю протоколу HTTP є можливість вказати в запиті і відповіді спосіб представлення одного і того ж ресурсу по різних параметрах: формат кодування, мова і т.д. Саме завдяки можливості вказівки способу кодування повідомлення клієнт і сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим.

CGI був спеціально розроблений для розширення можливостей WWW за рахунок підключення різноманітного програмного забезпечення.

CGI – це стандарт інтерфейсу, який використовується для організації взаємодії програми веб-сервера із зовнішньою програмою. Програму, яка працює по такому інтерфейсу спільно з веб-сервером, прийнято називати шлюзом. Також вживаються терміни "скрипт" (сценарій) або "CGI-програма". Сам інтерфейс розроблений таким чином, щоб можна було використовувати будь-яку мову програмування. Для обміну даними використовуються стандартні інтерфейси вводу/виводу.

**Веб-форум** – це клас веб-додатків для організації спілкування відвідувачів веб-сайту. Форум пропонує набір розділів для обговорення. Робота форуму заключається в створенні користувачами тем в розділах і наступним обговоренням всередині цих тем. Структура форуму: розділи – теми – повідомлення.

Форум відрізняється від чату розділенням обговорюваних тем і можливістю спілкування не в реальному часі.

На сьогодні форуми практично витіснили групи новин на базі NNTP і є одним з найбільш популярних способів обговорення питань в WWW.

**Блог** – це веб-сайт, головний змістом якого є записи, зображення чи мультимедіа, що регулярно додаються. Для блогів характерні короткі записи (пости) тимчасової значущості, відсортовані у зворотному хронологічному порядку (останній запис зверху). Звичайно, окремий пост у блозі має заголовок, дату публікації, власне, зміст, який складається з гіпертексту (думки автора, цитати, тощо), посилань на інші сайти та блоги в Інтернеті, інколи зображень чи навіть відео. Також пост містить коментарі до нього, залишені відвідувачами та просту веб-форму, за допомогою якої ці коментарі можна додати.

За версією газети "Washington Profile" першим блогом вважають сторінку Тіма Бернса-Лі, де він, починаючи з 1992 року, публікував новини. Широке використання блогів почалось з 1996 року. В серпні 1999 року комп'ютерна компанія "Pyra Labs" із Сан-Франциско відкрила сайт Blogger. Це була перша безкоштовна блогова служба. Згодом Blogger був викуплений компанією Google.

Інтернет-магазин – це інтерактивний веб-сайт, який рекламує товар або послугу, приймає замовлення на купівлю, пропонує користувачу вибір варіанту розрахунку, способу отримання замовлення та ін.

Інтернет-магазини створюються за допомогою систем управління контентом сайтів, які оснащені відповідними модулями. Крупні інтернет-магазини працюють на спеціально для них розроблених або адаптованих типових системах управління. Середні та малі магазини звичайно використовують типове комерційне та вільне ПЗ.

Інтернет-аукціон – аукціон, який проводиться з допомогою Інтернет. Велика перевага інтернет-аукціонів – доступність. Найпопулярнішим інтернет-аукціоном у світі є eBay.com, заснований у 1995 р.

### ***Запитання для самоконтролю***

- 1. Що називається службою (сервісом) Інтернет?*
- 2. Описати принцип роботи типу "клієнт-сервер".*

3. *Перерахувати відомі Вам служби Інтернету та коротко описати їх призначення.*
4. *Які є класифікації служб Інтернету? Навести приклади відповідних служб згідно класифікацій.*
5. *Як розшифровується WWW?*
6. *Що таке WWW?*
7. *На яких засадах базується служба WWW? Розшифрувати їх.*
8. *Чим відрізняється URL від URN та URI?*
9. *Що таке веб-форум? Принцип його роботи.*
10. *Що називають блогом? Як влаштований блог?*
11. *Перерахувати відомі Вам браузерери.*

## **Тема 6**

### **Пошук в Інтернет. Складові та функціонування пошукових систем**

**Пошукова система** – це служба (програмно-апаратний комплекс з веб-інтерфейсом), що надає можливість пошуку інформації в Інтернет (веб-сайт, на якому розміщено інтерфейс системи). Програмною частиною пошукової системи є **пошукова машина** – комплекс програм, що забезпечує функціональність пошукової системи і, зазвичай, є комерційною таємницею компанії-розробника пошукової системи.

Одним з перших способів організації доступу до інформаційних ресурсів мережі стало створення каталогів сайтів, в яких посилання на ресурси групувалися згідно тематики. Першим таким проектом став сайт Yahoo, що відкрився в квітні 1994 року. Після того, як число сайтів в каталозі Yahoo значно збільшилося, була додана можливість пошуку інформації по каталогу. Це не було пошуковою системою в повному розумінні, оскільки область пошуку була обмежена тільки ресурсами, присутніми в каталозі, а не всіма ресурсами мережі Інтернет.

Каталоги посилань широко використовувалися раніше, але практично втратили свою популярність в даний час. Причина: навіть сучасні каталоги, що містять величезну кількість ресурсів, представляють інформацію лише про дуже малу частку мережі Інтернет. Найбільший каталог мережі DMOZ (або Open Directory Project) містить інформацію про 5 мільйонів ресурсів, тоді як база пошукової системи Google складається із понад 8 мільярдів документів.

Першою повноцінною пошуковою системою став проект WebCrawler, що з'явився в 1994 році.

В 1995 році з'явилися пошукові системи Lycos і AltaVista. Остання довгі роки була лідером в області пошуку інформації в Інтернет.

В 1997 році Сергій Брін і Ларрі Пейдж створили Google в рамках дослідницького проекту в Стенфордському університеті. Зараз Google найпопулярніша пошукова система в світі.

23 вересня 1997 року була офіційно анонсована пошукова система Yandex, найпопулярніша в російськомовній частині Інтернет.

На даний час існує 3 основні міжнародні пошукові системи – Google, Yahoo і MSN Search, які мають власні бази і алгоритми пошуку. Більшість інших пошукових систем використовує в тому або іншому вигляді результати 3 вище перерахованих. Наприклад, пошук AOL (search.aol.com) і Mail.ru використовують базу Google, а AltaVista, Lycos і AllTheWeb – базу Yahoo.

**За принципом дії розрізняють кілька видів пошукових систем:**

- пошукові каталоги;
- рейтингові системи;
- індексні бази даних.

**Пошукові тематичні каталоги (Subject Directory)** організовані за тим самим принципом, що й тематичні каталоги бібліотек. На основній сторінці пошукового каталога розташовано скорочений список великих тематичних категорій, наприклад таких, як Освіта (Education), Наука (Science), Бізнес (Business), Мистецтво (Art) тощо.

Розрізняють пошукові каталоги **загального призначення** та **спеціалізовані**.

**Рейтингова система.** Ідея системи полягає у тому, що на головній сторінці розміщується перелік найважливіших тем, які, в свою чергу, містять список тематичних розділів. Вибравши потрібну тему, користувач отримує обмежений перелік веб-ресурсів, присвячених даній темі. Поруч з кожним веб-ресурсом висвітлюється оцінка його популярності, яка вимірюється в кількості переглядів цього ресурсу за останню добу. Проте, потрібно пам'ятати, що не завжди високий рейтинг об'єктивно відповідає якості веб-ресурсу.

**Індексні пошукові системи.** Більшість пошукових систем світу – індексні пошукові системи, які ще називають пошуковими покажчиками, пошуковими серверами, словниковими пошуковими системами, автоматичними індексами, пошуковими машинами, Search Engines. Їхнє призначення – якнайкраще охопити інформаційний веб-простір і подати його користувачам у зручному вигляді.

Принцип роботи з індексними пошуковими системами (ІПС) засновано на ключових словах. Розпочинаючи пошук інформації з певної теми, користувач має ввести ключові слова у рядок пошуку, які описують його тему і вибрати кнопку Знайти (або Найти, Go, Search).

Після цього ключова фраза посилається на сервер і пошукова система починає пошук у своїх базах даних адреси веб-ресурсів, які містять вказані ключові слова. Як здійснюється пошук, користувач не може бачити, він лише побачить результат запиту – нове вікно, в якому відображається перелік веб-ресурсів, які містять ключові слова.

Пошукова система складається з наступних основних компонентів:

**Spider (павук)** – браузероподібна програма, яка викачує веб-сторінки.

**Crawler (краулер, "мандрівний павук")** – програма, яка автоматично проходить по всіх посиланнях, знайдених на сторінці.

**Indexer (індексатор)** – програма, яка аналізує веб-сторінки, що завантажили павуки.

**Database (база даних)** – сховище завантажених і оброблених сторінок.

**Search engine results engine (система видачі результатів)** – витягує результати пошуку з бази даних.

**Web server (веб-сервер)** – веб-сервер, який здійснює взаємодію між користувачем і рештою компонентів пошукової системи.

**Spider** – це програма, яка завантажує веб-сторінки тим же способом, що і браузер користувача. Відмінність полягає в тому, що браузер відображає інформацію, що міститься на сторінці (текстову, графічну і т.д.), а павук не має ніяких візуальних компонент і працює напряму з html-текстом сторінки.

**Crawler** виділяє всі посилання, присутні на сторінці. Його завдання – визначити, куди далі повинен йти павук, ґрунтуючись на посиланнях або виходячи з наперед заданого списку адрес. Краулер, слідуючи по знайдених посиланнях, здійснює пошук нових документів, ще невідомих пошуковій системі.

**Індексатор** розбирає сторінку на складові частини і аналізує їх. Виділяються і аналізуються різні елементи сторінки, такі як текст, заголовки, структурні і стильові особливості, спеціальні службові html-теги і т.д. Деякі, наприклад, можуть індексувати кожне окреме слово у документі, інші – тільки 100 найбільш важливих слів в кожному документі. Може індексуватися розмір документу і кількість слів в ньому. Вигляд побудованого індексу визначає, який пошук може бути проведений пошуковим механізмом і як отримана інформація буде інтерпретована.

Перш, ніж сайт з'явиться в результатах пошуку, він повинен бути проіндексованим пошуковою системою. Індксація означає, що пошуковий робот відвідав сайт, проаналізував його і заніс інформацію в базу даних пошукової системи.

Якщо деяка сторінка занесена в індекс пошуковика, то вона може бути показана в результатах пошуку. Якщо сторінка в індексі відсутня, то пошукова система нічого не знає про неї, і, отже, ніяк не може використовувати інформацію з цієї сторінки.



**База даних** – це сховище всіх даних, які пошукова система завантажує і аналізує. Іноді базу даних називають індексом пошукової системи.

**Система видачі результатів** займається ранжуванням сторінок. Вона вирішує, які сторінки задовольняють запиту користувача, і в якому порядку вони повинні бути відсортовані. Це відбувається згідно алгоритмам ранжування пошукової системи.

В ідеальному випадку, розташованими першими в списку будуть документи, що є найбільш релевантними до запиту користувача. **Релевантність** – це відповідність змісту знайденої сторінки до запиту користувача. Пошукові системи використовують спеціальні алгоритми для визначення релевантності. Теоретичних методів визначення релевантності більш ніж 20. Але виділяють два основні напрями: лінгвістичне (Рамблер, Яндекс) і статистичне (Google).

Різні пошукові системи використовують різні алгоритми ранжування, однак основними принципами визначення релевантності є наступні:

- кількість слів запиту у html-кодi;
- теги, в яких ці слова розташовуються;
- місцезнаходження шуканих слів у документі;
- питома вага слів, відносно яких визначається релевантність, у загальній кількості слів документу;
- час – як довго сторінка знаходиться в базі пошукового сервера;
- індекс цитованості – як багато посилань на дану сторінку веде з інших сторінок, що зареєстровані у базі пошуковика.

Як правило, на **веб-сервері** присутня html-сторінка з полем введення, в якому користувач може задати фразу для пошуку. Веб-сервер також відповідає за видачу результатів користувачу у вигляді html-сторінки. Різні пошукові системи вибирають різні способи показу отриманого переліку результатів – деякі відображають лише посилання, інші виводять посилання з декількома першими реченнями документу або заголовком документу разом з посиланням. Коли користувач звертається до посилання на один з документів, цей документ завантажується з сервера, на якому він знаходиться.

Ю.Ліфшиць (<http://yury.name/modern/05modernnote.pdf>) описує структуру інформаційно-пошукової системи Google наступним чином (рис. 3):

URL Server – список усіх адрес;

Crawler – робот, який завантажує сторінки зі списку адрес і передає в Store Server;

Store Server зберігає сторінки в Repository, найчастіше у вигляді HTML-документа. При цьому вся додаткова інформація, така як картинки, flash-анімація та інше, не зберігається;

Indexer розбирає збережені в Repository HTML-документи у послідовності слів і зберігає їх у Barrels (база даних);

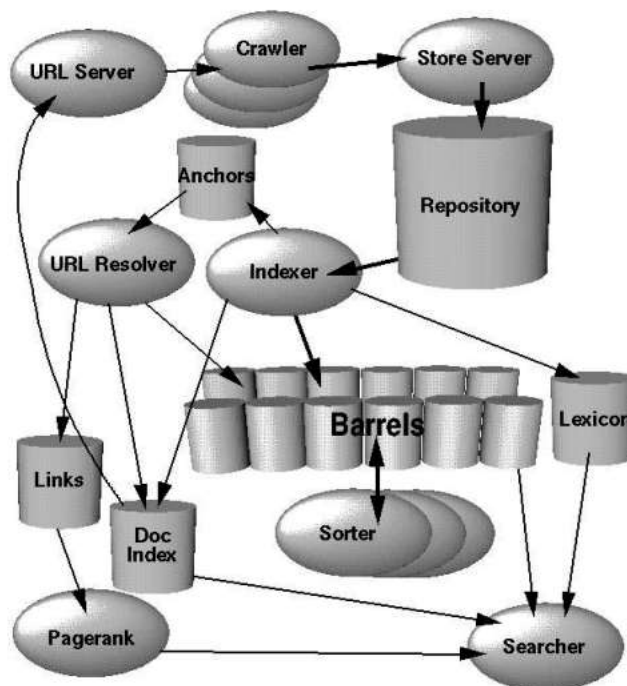


Рисунок 3 - Структура інформаційно-пошукової системи Google

Lexicon – список усіх слів. Найчастіше слова зберігаються в таблиці з двома полями "номер" і "слово". Таким чином досягається економія місця в базі даних, оскільки довгі слова замінюються досить коротким номером;

Anchors – виділені компонентом Indexer посилання (URL);

URL Resolver – обробник URL. Якщо знаходяться нові посилання, то вони передаються в URL Server;

Links визначає, які сайти на які посилаються, і передає це в PageRank;

PageRank – визначає рейтинг сайту, основним критерієм є кількість посилань на цей сайт;

Searcher – клієнт. Найчастіше клієнт користується статичною базою даних, яка оновлюється приблизно раз на добу.

### ***Запитання для самоконтролю***

1. *Що таке пошукова система?*
2. *Що називається пошуковою машиною?*
3. *Як поділяють пошукові системи за принципом дії? Охарактеризувати кожен із видів.*
4. *З яких частин складається пошукова система?*
5. *Для чого призначений spider (навук)?*
6. *Для чого призначений crawler (краулер)?*
7. *Які функції виконує Indexer (індексатор)?*
8. *Для чого призначена Search engine results engine (система видачі результатів)?*
9. *Що таке релевантність?*

## **Тема 7**

### **Сервіси мережі Інтернет: електронна пошта, сервіс FTP, файлообмінні мережі**

**Електронна пошта** – типовий сервіс відкладеного зчитування (off-line). Після відправлення повідомлення, як правило, у вигляді звичайного тексту, адресат отримує його на свій комп'ютер через деякий період часу, і знайомиться з ним, коли йому буде зручно.

Функціонування електронної пошти побудовано на принципі клієнт-сервер, стандартному для більшості мережевих сервісів. Щоб обмінюватись кореспонденцією з поштовим сервером, потрібно мати спеціальну програму-клієнт або веб-інтерфейс, який забезпечує доступ до електронної скриньки на

безкоштовному поштовому сервері. Незважаючи на різні засоби роботи з електронною поштою, можна виділити наступні загальні функції, які вони забезпечують:

- підготовка тексту;
- імпорт файлів-додатків; експорт файлів-додатків;
- відправка листа;
- перегляд і збереження кореспонденції; знищення кореспонденції;
- підготовка відповіді;
- коментування і пересилка інформації;
- ідентифікація власника поштової скриньки;
- автоматичне приєднання підпису;
- адресні книги;
- перевірка орфографії;
- можливість створення поштових скриньок різного призначення;
- робота у автономному (off-line) режимі;
- автоматична відправка відповіді про тимчасову відсутність отримувача

або автоматична переадресація листа на іншу адресу електронної пошти.

В Інтернеті для роботи з електронною поштою використовуються протоколи: SMTP (для надсилання повідомлень), POP та IMAP (для одержання).

На початковому етапі розвитку електронної пошти було цілком достатньо передавати між користувачами текстові повідомлення, що містили тільки символи з набору ASCII. Якщо користувачу потрібно було відправити зображення або виконуваний файл, він кодував його символами ASCII і вставляв у текст повідомлення. Із зростанням популярності електронної пошти і розширенням можливостей комп'ютерів і ліній передачі, з'явилася необхідність передавати поштою тексти, написані різними національними мовами, зображення, виконувані файли, відео та інші бінарні дані в уніфікованому форматі.

Стара схема передавання таких даних ставала все більш незручною через те, що:

- різні клієнти працювали з різними типами кодування, що часто призводило до неможливості прочитати адресатом отримані дані;

- не була визначена структура розміщення й ідентифікації типу кодованих даних. Щоб зрозуміти, що являє собою отримана інформація, її треба було вийняти з повідомлення і розкодувати.

Протокол SMTP – Simple Mail Transfer Protocol – був розроблений для обміну поштовими повідомленнями в мережі Internet. SMTP не залежить від транспортного середовища і може використовуватися для доставки пошти в мережах з протоколами, відмінними від TCP/IP.

Основне завдання протоколу SMTP полягає в тому, щоб забезпечувати передачу електронних повідомлень (пошту). Для роботи через протокол SMTP клієнт створює TCP з'єднання з сервером. Потім клієнт і SMTP-сервер обмінюються інформацією поки з'єднання не буде закрито або перервано. Основною процедурою в SMTP є передача пошти (Mail Procedure), а також перевірка імен поштової скриньки і виведення списків поштових груп. Найпершою процедурою є відкриття каналу передачі, а останньою – його закриття. Команди SMTP вказують серверу, яку операцію хоче зробити клієнт.

Протокол був розроблений для передачі тільки тексту в кодуванні ASCII. Це не дає можливості відсилати текст на національних мовах (наприклад, кирилиці), а також відправляти двійкові файли (такі як зображення, відеофайли, програми або архіви). Для зняття цього обмеження був розроблений стандарт MIME, який описує спосіб перетворення двійкових файлів в текстові.

Стандарт MIME – Multipurpose Internet Mail Extensions (Багатоцільове розширення електронної пошти) або RFC-1341 – призначений для опису тіла поштового повідомлення Інтернет, був розроблений для більш зручної роботи зі складними і нестандартними (не ASCII) повідомленнями. Цей стандарт описує передачу різних типів даних по електронній пошті, а також є специфікацією для кодування інформації і форматування повідомлень таким чином, щоб їх можна було пересилати через Інтернет.

MIME визначає механізми для передачі різного роду інформації всередині текстових даних (зокрема, за допомогою електронної пошти), а саме: текст на мовах, для яких використовуються кодування, відмінні від ASCII, і нетекстовий контент, такий як картинки, музика, фільми, програми.

Протокол **POP** – Post Office Protocol – Протокол поштового відділення – призначений для розбору пошти з поштових скриньок користувачів на їх робочі місця за допомогою програм-клієнтів. Якщо по протоколу SMTP користувачі відправляють кореспонденцію через Internet, то по протоколу POP користувачі одержують кореспонденцію зі своїх поштових скриньок на поштовому сервері в локальні файли. Стандарт протоколу POP3 визначений в RFC 1939.

POP-сервер не відповідає за відправлення пошти, він працює тільки як універсальна поштова скринька для групи користувачів. Як правило, під час роботи з електронною поштою невеликі організації використовують для одержання своєї кореспонденції POP3-сервер, встановлений на будь-якому комп'ютері в офісі, а відправляють пошту по SMTP на один із добре доступних загальновідомих SMTP-серверів міста.

Протокол **IMAP** – Internet Message Access Protocol – Протокол доступу до електронної пошти Internet – дозволяє клієнтам одержувати доступ і маніпулювати повідомленнями електронної пошти на сервері.

Протокол IMAP4 відрізняється від протоколу POP3 тим, що IMAP4 підтримує роботу із системою каталогів (або папок) повідомлень. IMAP4 дозволяє керувати каталогами (папками) віддалених повідомлень так само, якби вони розташовувалися на локальному комп'ютері. IMAP4 дозволяє клієнту створювати, видаляти та перейменовувати поштові скриньки, перевіряти наявність нових повідомлень і видаляти старі. Завдяки тому, що IMAP4 підтримує механізм унікальної ідентифікації кожного повідомлення в поштовій папці клієнта, він дозволяє читати з поштової скриньки тільки повідомлення, що задовольняють визначені умови або їх частини, змінювати атрибути повідомлень і переміщати окремі повідомлення. Структура папок значною мірою залежить

від типу поштової системи, але в будь-якій системі клієнт має спеціальний каталог INBOX, до якого потрапляють повідомлення, що надійшли клієнту.

Сервіс FTP – це один з найстаріших сервісів Інтернет. FTP – File Transfer Protocol – протокол передачі файлів, але при цьому мається на увазі не просто протокол, а саме сервіс – доступ до файлів у файлових архівах.

Технологію FTP було розроблено в рамках проекту ARPA. Вона призначена для обміну великими об'ємами інформації між машинами з різною архітектурою і її підтримують всі популярні операційні системи. Основу технології складає FTP-протокол.

FTP-протокол – один із базових протоколів передачі даних, що з'явився значно раніше за протокол HTTP. Перші специфікації FTP відносяться до 1971 року. Даний протокол дозволяє підключатися до серверів FTP, переглядати вміст каталогів, завантажувати файли з сервера на локальний комп'ютер і навпаки. В рамках служби FTP адміністратор може створювати на сервері FTP-архіви і визначати спосіб взаємодії користувачів з цією службою.

Доступ до файлів за протоколом FTP відбувається за допомогою FTP-клієнтів, які здійснюють з'єднання з FTP-сервером.

FTP-сервер – комп'ютер, на якому запущено відповідне програмне забезпечення, що надає доступ до файлів і каталогів цього комп'ютера за протоколом FTP. FTP-сервери можуть бути загальнодоступними (anonymous FTP-site) і з обмеженим доступом. На FTP-серверах організовано FTP-архіви – спеціальні дерева каталогів, доступ до яких надається віддаленому користувачу.

FTP – сервіс прямого доступу, який вимагає повноцінного підключення до Інтернет, але є можливість доступу і через електронну пошту. FTP-сервіси, на відміну від електронної пошти, практично не збільшують розмір файлів, які передаються, можуть передавати файли великого розміру, крім того, FTP-сервіси дозволяють "докачувати" файли у разі розриву зв'язку, відновлюючи передачу файлу, при відновленні зв'язку, з перерваного місця.

Щоб скопіювати файл із серверу FTP на комп'ютер користувача, необхідно спочатку задати адресу відповідного FTP-серверу, а потім зареєструватись –

ввести своє ім'я та пароль. Якщо користувач не має на сервері свого облікового запису (zareestrovanoго імені та пароля), він може зайти лише на загальнодоступні сервери (режим анонімності). На деякі з таких серверів можна потрапити автоматично, інші вимагають реєстрації анонімного користувача. Тоді на запит імені необхідно ввести – anonymous, а на запит пароля – будь-яке слово.

Деякі файли на FTP-серверах можуть бути доступні кожному, інші – лише окремим користувачам, які мають на це право доступу. Загальнодоступні файли, зазвичай, розташовані в папці pub (public – загальнодоступний).

Служба FTP базується на двох стандартах: URL та FTP – File Transfer Protocol – протокол передачі файлів.

**Протокол передачі файлів** FTP дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі. Встановивши зв'язок із віддаленим комп'ютером, користувач може скопіювати файл із віддаленого комп'ютера на свій або скопіювати файл із свого комп'ютера на віддалений. Незважаючи на те, що передача файлів від однієї системи до іншої здається простою задачею, спочатку повинні бути розв'язані деякі проблеми. Наприклад, дві системи можуть використовувати різні домовленості про імена файлів. Дві системи можуть мати різні шляхи для представлення текстів і даних. Дві системи можуть мати різні структури директорій. Всі ці проблеми вирішує FTP.

Протокол FTP не призначений для розсилання файлів кільком користувачам. Він призначений лише для обміну файлами між окремими комп'ютерами. У ньому застосовують клієнт-серверну систему.

FTP відрізняється від інших додатків типу клієнт-сервер тим, що він встановлює два з'єднання між хостами. Одне з'єднання застосовується для передачі даних, інше – для управління інформацією (команди і відгуки), що робить FTP більш ефективним.

**Файлові сховища** або **файловий хостинг** – сервіс, що надає користувачу місце під його файли і цілодобовий доступ до них через веб-інтерфейс, як правило за протоколом HTTP. Такий сервіс дозволяє зручно "обмінюватися"



файлами. На спеціальній сторінці файлового сховища (найчастіше на головній) користувач завантажує файл на сервер сховища і отримує постійне посилання, яке він може розсилати по e-mail, публікувати в блогах, на форумах тощо. Перейшовши за таким посиланням, інший користувач може викачати розміщений файл.

Завдяки файловим хостингам вдалося вирішити проблему перевантаження для ряду ресурсів, а також надати для звичайного користувача можливість передавання файлу, наприклад, розміром 100 мБ безкоштовно.

Популярні файлові сховища: <http://rapidshare.com>; <http://ifolder.ru>; <http://depositfiles.com>; <http://letitbit.net>; <http://webfile.ru>; <http://fileshare.in.ua>.

**Файлообмінні мережі.** Термін файлообмінні мережі стосується сукупної назви однорангових мереж для сумісного використання файлів (P2P). Тобто таких мереж, які базуються на рівноправності учасників обміну файлами (кожен учасник одночасно є клієнтом і сервером).

#### **Функціонування однорангових мереж.**

Кожна із машин однорангової мережі може зв'язуватися з іншими. Кожна з них може надсилати запити іншим машинам цієї мережі для надання їй певних ресурсів (виступати в ролі клієнта). Будучи сервером, кожна машина повинна мати можливість обробляти запити від інших машин в мережі. Кожна машина також повинна виконувати деякі допоміжні та адміністративні функції (наприклад, зберігати список інших відомих машин-"сусідів").

Кожен учасник даної мережі не гарантує свою присутність на постійній основі. Він може з'являтися та зникати в довільний момент часу. Але при досягненні певного критичного розміру мережі настає такий момент, коли в мережі одночасно існує багато серверів з однаковими функціями.

Вперше термін peer-to-peer (P2P) був використаний в 1984 році компанією IBM при розробці мережевої архітектури для динамічної маршрутизації трафіка через комп'ютерні мережі з довільною топологією (Advanced Peer to Peer Networking). Сьогодні на технології P2P базується велика кількість популярних мережевих сервісів.

В основі технології лежить принцип децентралізації: всі вузли в мережі P2P рівноправні, що забезпечує такі переваги над клієнт-серверним підходом:

- збільшення швидкості отримання даних зарахунок копіювання одночасно з декількох джерел;
- можливість розподілення ресурсів без "прив'язки" до конкретних IP-адрес;
- величезна потужність мережі в цілому та ін.

Основні **принципи роботи файлообміну** полягають в наступному:

- користувач викачує програму до себе на комп'ютер;
- дає доступ іншим користувачам до частини своїх ресурсів;
- наявний засіб пошуку шукає ресурси, викладені на комп'ютерах інших користувачів для вільного викачування.

Будь-який користувач, використовуючи пошук, може знайти на комп'ютері будь-якого іншого користувача ті ресурси, які він виклав у вільний доступ, і безплатно викачати їх. А оскільки кількість користувачів таких файлообмінних програм обчислюється сотнями тисяч, а іноді навіть мільйонами, користувач напевно знайде файл, що цікавить його.

За деякими даними, на сьогодні більше половини трафіка припадає на трафік файлообмінних P2P-мереж.

#### **Типи файлообмінних мереж:**

- централізовані;
- децентралізовані;
- частково централізовані (гібридні).

#### **Централізовані мережі**

У централізованих файлообмінних мережах службову інформацію зберігають сервери індексацій. Перевагою такої мережі є відносна простота її програмування та невеликий об'єм службової інформації, що передається на сервери. Недоліком є те, що сервери індексацій можуть бути відключені з різних причин (технічний збій, розорення компанії, що ним володіє, судове рішення). У такому разі мережа перестає функціонувати.

## **Децентралізовані мережі**

Децентралізовані файлообмінні мережі функціонують без серверів індексацій. Незважаючи на те, що об'єм службової інформації, що передається в них більший, надійність їх набагато вища. Поки не існує способу насильно зупинити функціонування такої мережі.

## **Частково децентралізовані (гібридні) мережі**

Поєднуючи швидкість централізованих мереж і надійність децентралізованих, використовуються гібридні схеми, в яких використовуються незалежні індексації сервери, що постійно синхронізують інформацію між собою. Таким чином, при виході з ладу одного з них, мережа продовжує функціонувати.

Переваги технології P2P перед клієнт-серверним підходом:

- відмовостійкість при втраті зв'язку з кількома вузлами мережі;
- збільшення швидкості отримання даних за рахунок копіювання одночасно з кількох джерел;
- можливість розділення ресурсів без "прив'язки" до конкретних IP-адрес;
- величезна потужність мережі в цілому тощо.

P2P-технологія також застосовується у:

- телебаченні і аудіотрансляції;
- паралельному програмуванні;
- розсилці повідомлень і статей;
- підтримці системи доменних імен;
- резервному копіюванні і створенні стійких розподілених сховищ даних та ін.

Серед файлообмінних мереж за кількістю вузлів лідерами є Bittorent, eDonkey, Gnutella2, Gnutella2.

## ***Запитання для самоконтролю***

1. *Описати поняття електронної пошти.*
2. *Перерахувати основні функції, які забезпечують поштові клієнти.*

3. *Які протоколи використовуються для роботи з електронною поштою?*
4. *Для чого призначений стандарт MIME?*
5. *Перерахувати відомі програми – поштові клієнти.*
6. *Що таке служба FTP?*
7. *Що таке протокол FTP?*
8. *Що таке файлообмінна мережа?*
9. *Перерахувати основні принципи роботи файлообміну.*
10. *Що таке однорангова мережа?*
11. *Як функціонують однорангові мережі?*
12. *На які типи поділяються файлообмінні мережі?*
13. *Де застосовується технологія P2P?*
14. *Перерахувати відомі Вам файлообмінні мережі.*

## **Тема 8**

### **Сервіси Інтернет: соціальні мережі, служба миттєвих повідомлень, сервіси Google, RSS**

Взагалі кажучи, соціальна мережа – це соціальна структура, що складається з групи вузлів, якими є соціальні об'єкти (люди або організації), і зв'язків між ними.

Термін "соціальна мережа" був введений задовго до появи Інтернет і, власне, сучасних Інтернет-мереж в 1954 році соціологом з "Манчестерської школи" Джеймсом Барнсом.

Соціальна мережа в Інтернет – це інтернет-спільнота користувачів, об'єднаних за якоюсь ознакою на базі одного сайту, який і називається в цьому випадку соціальною мережею. Іншими словами, соціальна мережа у Інтернет будується на тих же принципах, що і в реальному світі, але відрізняється від

реальних людських спільнот тим, що у функціонуванні мережі не грає ролі географічна віддаленість її учасників один від одного.

В середині дев'яностих років ХХ століття з'явилися такі веб-інструменти, використання яких дозволяло встановлювати зв'язки між користувачами в рамках одного сайту. Спочатку користувачі об'єднувалися в мережі для підтримки ділових контактів. Пізніше з'явилися форуми за інтересами.

У другій половині ХХ століття соціальні мережі стали активно розвиватися як наукова концепція, спочатку вони стали популярні на заході. Потім це професійне поняття соціологів перетворилося на концепцію, яка є однією з центральних у концепції веб 2.0, яку як поняття вперше ввів Тім О'Рейлі 30 вересня 2005 в своїй статті "Tim O'Reilly – What Is Web 2.0".

За даними <http://vincos.it/2012/06/11/la-mappa-dei-social-network-nel-mondo-giugno-2012/> (червень, 2012), географічний розподіл популярних соціальних мереж виглядає наступним чином (рис. 4):

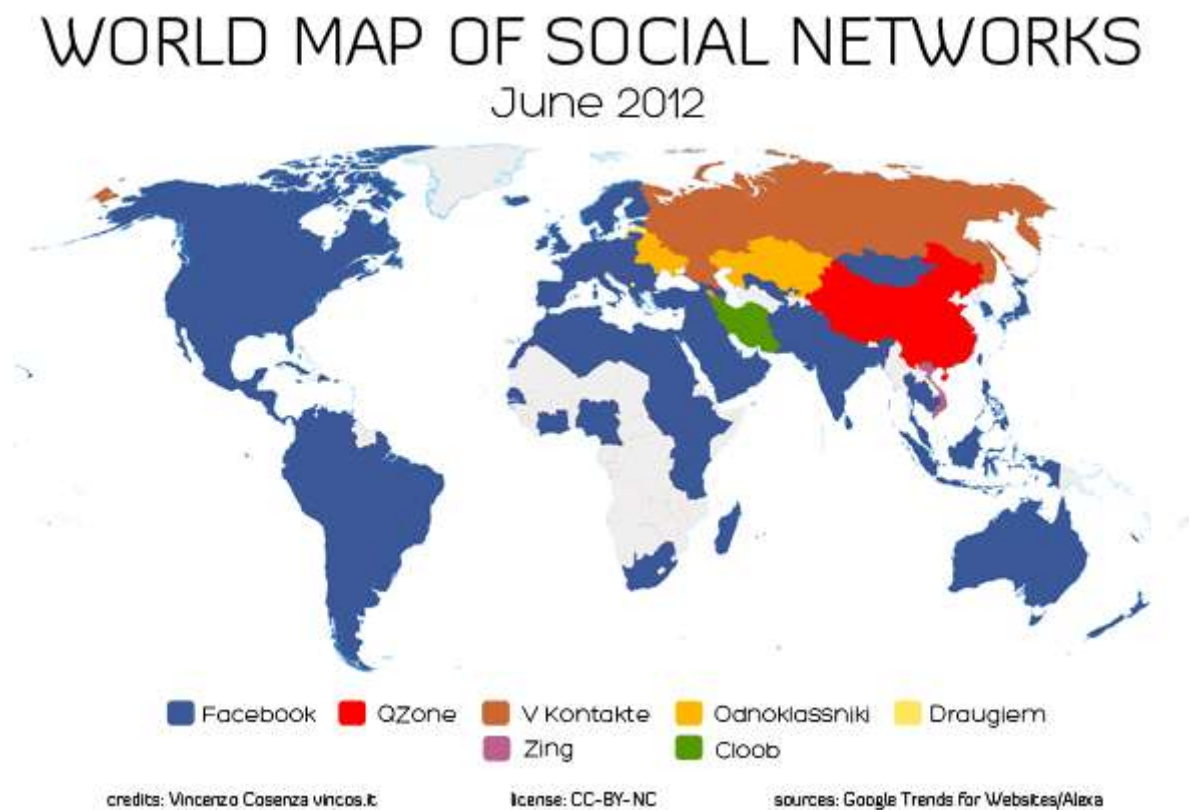


Рисунок 4 - Світова карта соціальних мереж

Система обміну повідомленнями є одним з найбільш доступних і затребуваних засобів спілкування в Інтернет, в корпоративних і локальних

мережах. Служби обміну повідомленнями поділяються на служби обміну повідомленнями в режимі off-line (поштові системи) та служби миттєвих повідомлень (Internet Relay Chat і Instant Messaging Service) у режимі on-line. Технологія миттєвого обміну повідомленнями забезпечує обмін текстовими повідомленнями в режимі реального часу з використанням спеціального програмного забезпечення через Інтернет. Ці служби базуються на мережевій архітектурі клієнт-сервер.

Служба IRC (Internet Relay Chat або Чат) є першим засобом для онлайн-спілкування, яка надає великий вибір каналів (тем) для проведення дискусій з однодумцями. Чат – це текстовий діалог у реальному часі.

Для спілкування в чаті можна використовувати як IRC-клієнти, так і веб-чати. Веб-чати призначені для обміну повідомленнями на сервері (веб-сторінці) за допомогою браузера, в цьому випадку встановлювати на ПК клієнтський додаток не потрібно. Веб-чат – це веб-сторінка, на якій в реальному часі можна спілкуватися з іншими відвідувачами.

Результатом розвитку чату стала служба миттєвих повідомлень (Instant Messaging Service, IMS). В IMS крім текстових повідомлень можна передавати, звукові сигнали, картинки, відео, файли. Мережева архітектура IMS побудована за принципом клієнт-сервер. Ця служба має свої мережі. Клієнтська програма IMS, яка призначена для ведення бесіди і миттєвого обміну повідомленнями в режимі онлайн через служби миттєвих повідомлень, називається месенджером (Instant messengers, IM).

Більшість мереж служби миттєвих повідомлень використовують закриті або пропріетарні протоколи (власні протоколи, що належать тільки однієї мережі) обміну інформацією. В основному в кожній з таких мереж застосовується свій месенджер.

Між різними мережами IMS зазвичай відсутні взаємозв'язку, тому месенджер однієї мережі, наприклад, ICQ не може зв'язатися з месенджером мережі Skype. Це означає, що для ведення спілкувань між собою користувачі повинні зареєструватися в одному і тому ж сервісі і встановити їх месенджери.

Але існують й альтернативні месенджери для служб миттєвих повідомлень, які можуть одночасно працювати в декількох мережах. Наприклад, безкоштовний відкритий багатопрокольний модульний клієнт (месенджер) Miranda IM (або Trillian, Pidgin) дозволяє підключатися одночасно до декількох мереж, що позбавляє від необхідності встановлювати окремий месенджер для кожної мережі.

Список основних функцій, які можуть надавати сучасні месенджери служб миттєвих повідомлень:

- чат (відеочат, текстовий і голосовий);
- VoIP сервіси: дзвінки на комп'ютер, дзвінки на стаціонарні і мобільні телефони;
- можливість відправлення SMS;
- передача файлів;
- інструменти для спільної роботи в режимі реального часу;
- можливість спілкуватися в чаті безпосередньо на веб-сторінці;
- нагадування і оповіщення;
- зберігання історії спілкування по кожному контакту;
- індикація про мережевий статус користувачів (в мережі, немає на місці і т.д.), занесених до списку контактів та ін.

"Google Inc." – американська корпорація, заснована 7 вересня 1998 року як приватна компанія, що займається розробкою, розвитком і дизайном найпопулярнішого в Інтернет пошукового сервісу. Google підтримує і розробляє ряд інтернет-сервісів і продукції (рис. 5).

Компанія була заснована Ларі Пейджем і Сергієм Бріном. Проголошеною місією компанії від самого початку була "організація світової інформації, забезпечення її доступності і користі для всіх".

Назва "Google" – спотворене написання слова "гугол" (googol), яке придумав Мілтон Сіротта, племінник американського математика Едварда

Кайзера. "Гугол" – це число  $10^{100}$ , десятковий запис його містить одиницю та сто нулів.

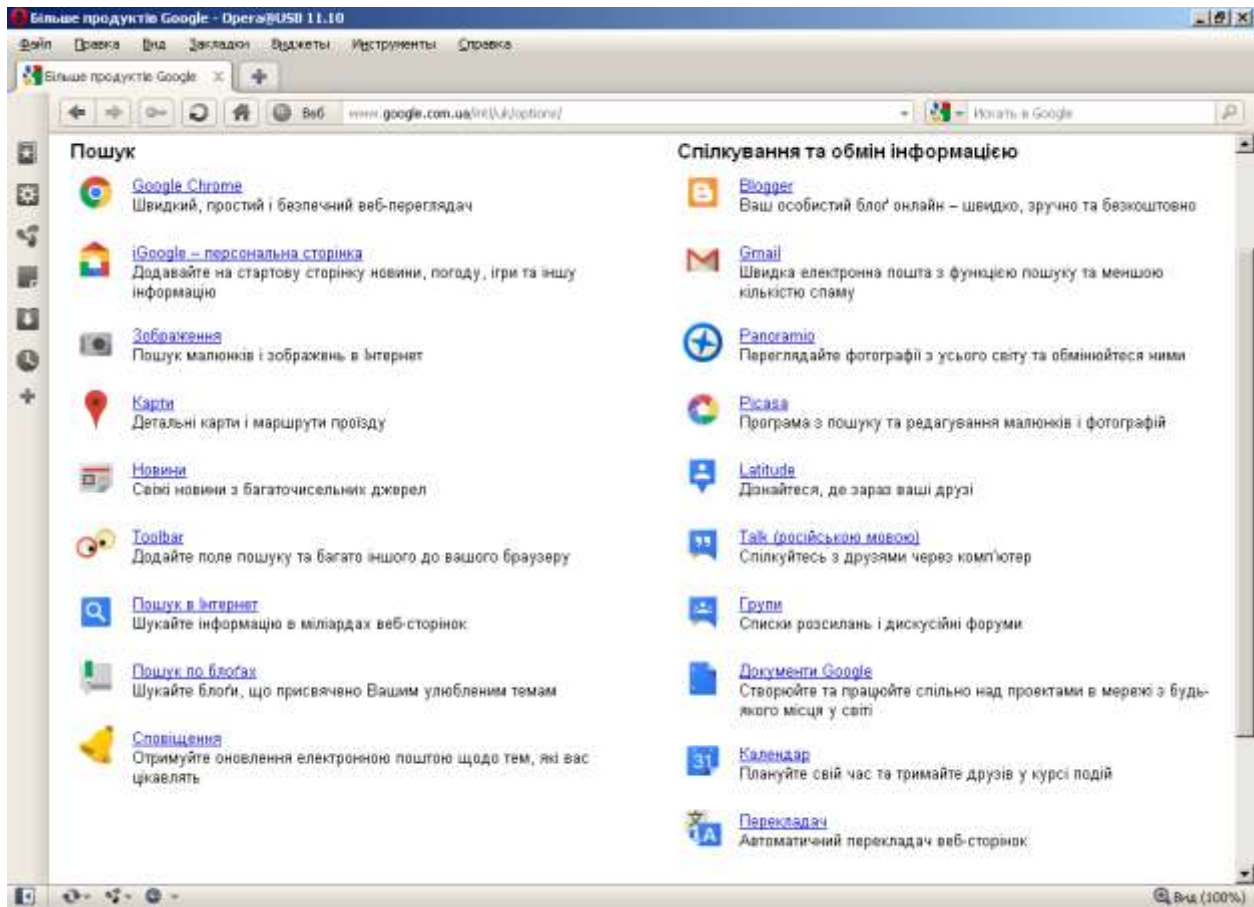


Рисунок 5 - Перелік сервісів Google

Ще однією технологією, яка набуває все більшої популярності в мережі Інтернет є RSS – Rich Site Summary. За висловлюванням популярного блогера Даррена Роуза (Darren Rowse): RSS – це технологія, яку використовують мільйони користувачів Інтернет з усього світу, щоб отримувати поновлення на своїх улюблених веб-сторінках. Офіційне ж визначення технології RSS наступне: це родина XML-форматів, що використовується для публікації та постачання часто змінюваної інформації, наприклад, нових записів в блозі, заголовків новин, анонсів статей, зображень, аудіо і відео матеріалів (в стандартизованому форматі). Документ в стандарті RSS (який також інколи називають "стрічкою", "веб-стрічкою" або "каналом") складається з повного, або часткового тексту і метаданих (дата і авторство).

Виділяють наступні версії RSS:

- Rich Site Summary (RSS 0.9x) – збагачена анотація сайту;



- RDF Site Summary (RSS 0.9 и 1.0) – анотація сайту з застосуванням інфраструктури опису ресурсів;
- Really Simple Syndication (RSS 2.x) – дуже простий збір зведеної інформації.

За даними <http://webempires.org/dominating-websites-map/> розподіл найвідвідуваніших сайтів у світі по регіонам виглядає наступним чином (рис. 6):

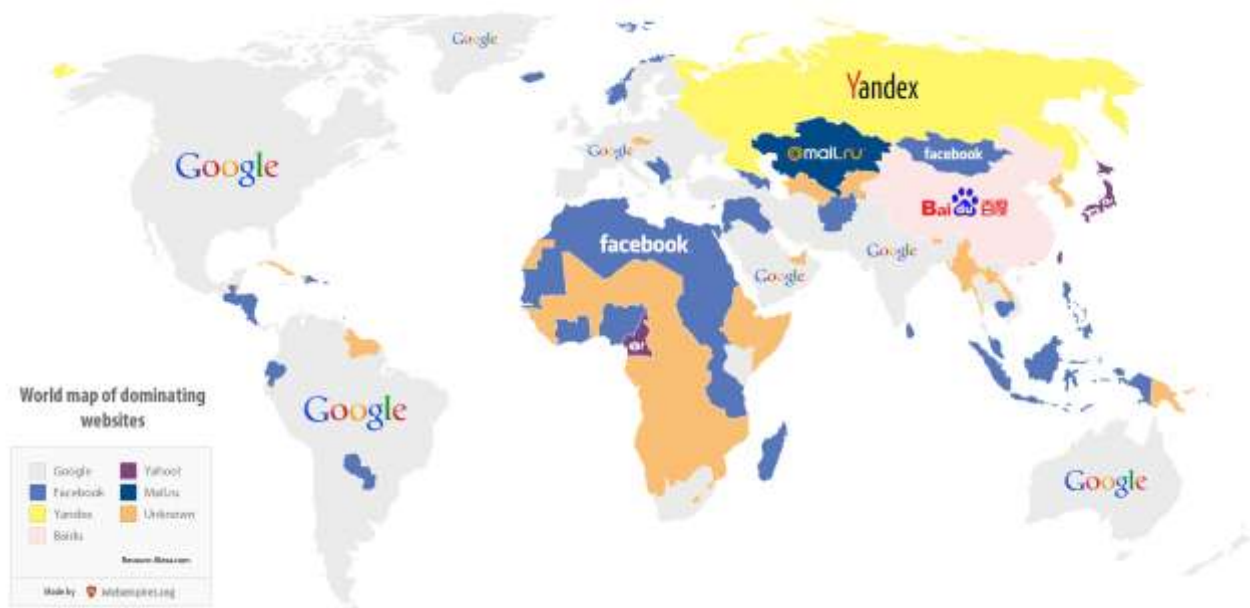


Рисунок 6 - Найбільш відвідувані сайти у світі

### ***Запитання для самоконтролю***

1. *Що таке соціальна мережа?*
2. *Перерахувати відомі Вам соціальні мережі.*
3. *Що таке RSS?*
4. *Що таке система обміну повідомленнями?*
5. *Яка різниця між IRC і IMS?*
6. *Як називається програмне забезпечення для обміну миттєвими повідомленнями? Перерахувати відоме Вам відповідне ПЗ.*
7. *Які функції надають месенджери?*
8. *Перерахувати відомі Вам служби Google, їх можливості та сфери застосування.*
9. *Що таке і для чого використовується технологія RSS?*

## Тема 9

### Мова гіпертекстової розмітки HTML. Структура HTML-документу. Робота з текстом в HTML

**HTML** (HyperText Markup Language – мова розмітки гіпертексту) – стандартна мова розмітки документів у WWW, яка визначає, як і які елементи повинні розміщуватися на веб-сторінці. Документ HTML обробляється браузером та відтворюється на екрані у звичному для людини вигляді.

Мова HTML була розроблена британським вченим Тімом Бернерсом-Лі приблизно в 1991-1992 роках в Європейському інституті ядерних досліджень в Женеві. HTML створювалась як мова для обміну науковою та технічною документацією, придатна для використання людьми, які не є спеціалістами в області верстки. HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту.

Тег – це елемент мови розмітки гіпертексту, призначений в основному для задання того, як буде відображатися текст (сторінка). Загальний синтаксис написання тегів:

**<тег атрибут1="значення" атрибут2="значення">**

**<тег атрибут1="значення" атрибут2="значення">...</тег>**

Отже, теги бувають двох типів: одинарні і парні (контейнери). Одинарний тег використовується самостійно, а парні можуть містити всередині себе інші теги (вкладені) або текст. Має значення порядок тегів. Закриваючі теги бувають обов'язковими, необов'язковими або можуть не використовуватись взагалі. При написанні тегів можна використовувати як великі так і малі літери. Вміст тегу можна відображати в декількох рядках.

Теги можуть мати атрибути (властивості), які розділяються між собою пробілами. Не всі теги мають атрибути. Умовно атрибути можна поділити на обов'язкові та необов'язкові. Згідно специфікації HTML всі значення атрибутів слід записувати в подвійних або одинарних лапках. Допускається використовувати деякі атрибути, не вказуючи для них значення.

Порядок атрибутів в тегах не має значення. При заданні значень атрибутів слід дотримуватися правильних типів (текст, число, шлях до файлу та ін.)

Невідомі теги і атрибути (або невірно написані) будуть браузером проігноровані і текст буде відображатися так, ніби тег не був написаний.

Веб-сторінку (HTML-документ) можна створювати в довільному текстовому редакторі. При цьому слід дотримуватися правильної структури написання документа і при збереженні файлу примусово вказати розширення .html або .htm. Для відображення змісту веб-сторінки у браузері, збережений файл відкривають у потрібному браузері.

### **Структура HTML-документа:**

**<html>**

**<head>**

**<title>**

**</title>**

**</head>**

**<body>**

**</body>**

**</html>**

**<html> ... </html>** – вказують на те, що веб-сторінка створена засобами HTML.

**<head> ... </head>** – початок і кінець заголовка документа. Крім назви документа в цей розділ може включатися службова інформація.

**<title> ... </title>** – все, що знаходиться між тегами **<title>** і **</title>**, сприймається браузером як назва документа. Відображається в рядку заголовка браузера, який використовується для перегляду веб-сторінки. Рекомендується назва не довше 64 символів. Даний тег записується тільки в розділі заголовку HTML-документа.

**<body> ... </body>** – початок і кінець тіла HTML-документа, яке визначає вміст документа.

Атрибути (параметри) тегу **<body>**:

**bgcolor** – встановлює колір фону документа;  
**text** – встановлює колір символів;  
**link** – колір, яким відображатимуться посилання на інші документи;  
**vlink** – цим кольором виділяються вже відвідані посилання;  
**alink** – колір для позначення посилання під час натиснення на них мишою;  
**background** – задає фонове зображення веб-сторінки.

Колір в цих параметрах задається в форматі: #RRGGBB, де R (red – червоний), G (green – зелений), B (blue – синій) – це складові компоненти кольору в шістнадцятірковій формі. Наприклад, якщо треба отримати зелений колір, для якого складові червоного і синього нульові, то потрібно записати #00FF00. Альтернативою є використання назв кольорів англійською мовою.

Заголовки задаються тегами `<h1> ... </h1>` – `<h6> ... </h6>`. Вони описують заголовки шести різних рівнів. Заголовок першого рівня – найкрупніший, шостого рівня – найдрібніший.

Для задання абзацу використовується тег `<p>`. Закриваючий тег необов'язковий. Вирівнювання тексту в абзаці задається за допомогою атрибуту **align**, який може приймати значення **left**, **right**, **center** і **justify** – вирівнювання за шириною. За замовчуванням – вирівнювання по лівому краю.

Параметри шрифту задаються тегом `<font>`. Закриваючий тег обов'язковий. Деякі атрибути тегу `<font>`:

**face** – задає назву шрифту, яким потрібно відобразити текст в документі.  
Наприклад: `<font face="Arial">Arial</font>`;

**size** – задає висоту символів в межах від 1 до 7.

Деякі теги для відображення символів:

`<sub> ... </sub>` – перетворення символів в нижній індекс;

`<sup> ... </sup>` – перетворення символів в верхній індекс;

`<b> ... </b>` – відображення символів напівжирним шрифтом;

`<i> ... </i>` – відображення символів курсивом (нахиленим) шрифтом;

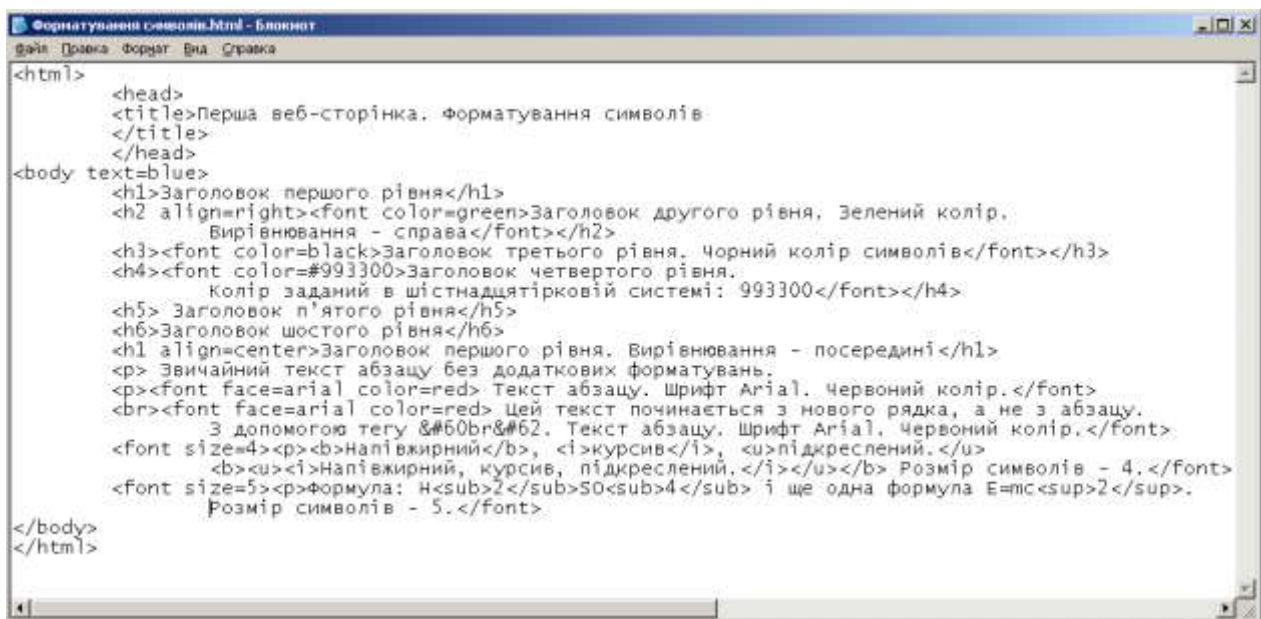
`<u> ... </u>` – відображення символів підкресленим шрифтом;

`<strike> ... </strike>` – відображення символів закресленими;

`<pre> ... </pre>` – текст відображається з постійною шириною символу, не пропускає пропуски, табуляції і перенесення рядків, залишаючи текст таким, яким він був відформатований раніше.

Тег `<br>` (від **break line**) забезпечує переведення курсора в наступний рядок без відступу від попереднього рядка (`<p>` переводить курсор в наступний рядок і дає великий відступ від попереднього рядка).

Наприклад, нехай у вікні текстового редактора "Блокнот" маємо код (рис. 7):



```
<html>
<head>
<title>Перша веб-сторінка. Форматування символів
</title>
</head>
<body text=blue>
<h1>Заголовок першого рівня</h1>
<h2 align=right<font color=green>Заголовок другого рівня, Зелений колір,
Вирівнювання - справа</font></h2>
<h3><font color=black>Заголовок третього рівня. Чорний колір символів</font></h3>
<h4><font color=#993300>Заголовок четвертого рівня.
Колір заданий в шістнадцятірковій системі: 993300</font></h4>
<h5> Заголовок п'ятого рівня</h5>
<h6>Заголовок шостого рівня</h6>
<h1 align=center>Заголовок першого рівня. Вирівнювання - посередині</h1>
<p> Звичайний текст абзацу без додаткових форматувань.
<p><font face=arial color=red>Текст абзацу. Шрифт Arial. Червоний колір.</font>
<br><font face=arial color=red>Цей текст починається з нового рядка, а не з абзацу.
З допомогою тегу &#50br&#62. Текст абзацу. Шрифт Arial. Червоний колір.</font>
<font size=4><p><b>Напівжирний</b>, <i>курсив</i>, <u>підкреслений.</u>
<b><u><i>Напівжирний, курсив, підкреслений.</i></u></b> Розмір символів - 4.</font>
<font size=5><p>Формула:  $H_2SO_4$  і ще одна формула  $E=mc^2$  і ще одна формула  $E=mc^2$ . Розмір символів - 5.</font>
</body>
</html>
```

Рисунок 7

Тоді у браузері веб-сторінка буде мати вигляд (рис. 8):

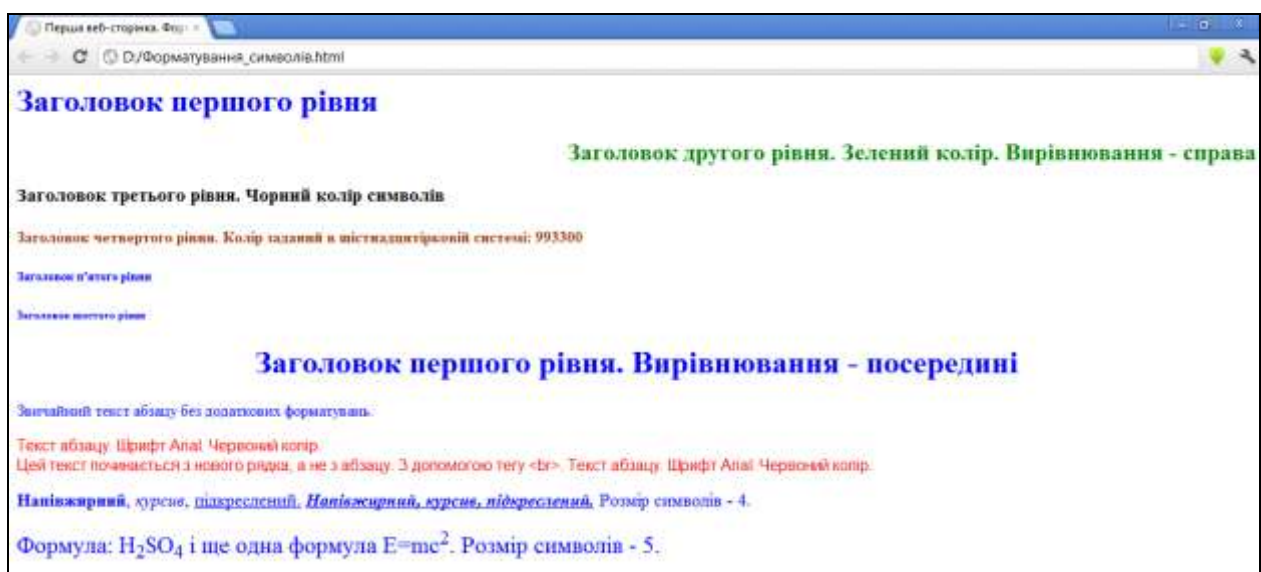


Рисунок 8

В HTML розрізняють три види списків:

- нумеровані (марковані);
- нумеровані;
- списки визначень.

#### **Синтаксис нумерованого списку:**

**<ul>**

**<li>** перший елемент списку

**<li>** другий елемент списку

**<li>** третій елемент списку

...

**</ul>**

За замовчуванням маркером нумерованого списку виступає круг чорного кольору. Для зміни маркера використовується атрибут **type**, який може приймати значення **disc** (чорний круг), **square** (квадрат) і **circle** (коло). Параметр **type** можна записувати як в тегу **<ul>** для задання певного маркера для всього списку, так і для тегу **<li>** для задання даного маркера.

#### **Синтаксис нумерованого списку:**

**<ol>**

**<li>** перший елемент списку

**<li>** другий елемент списку

**<li>** третій елемент списку

...

**</ol>**

Атрибут **type** задає тип нумерації:

**type=1** – тип нумерації арабськими цифрами: 1, 2, 3...;

**type=a** – тип нумерації латинськими літерами: a, b, c...;

**type=A** – тип нумерації латинськими прописними літерами: A, B, C...;

**type=i** – римські цифри нижнього регістру: i, ii, iii, ...;

**type=I** – римські цифри верхнього регістру I, II, III, IV...

**Списки визначень** використовуються для опису визначень і термінів.

Синтаксис:

**<dl>**

**<dt>**Термін 1**<dd>**Визначення першого терміну

**<dt>**Термін 2**<dd>**Визначення другого терміну

...

**</dl>**

### ***Запитання для самоконтролю***

1. *Що таке HTML?*
2. *Що таке тег?*
3. *Що таке веб-сторінка?*
4. *Чи є мова HTML мовою програмування?*
5. *Описати загальний синтаксис написання тегів.*
6. *Хто є розробником мови HTML?*
7. *Чи є мова HTML регістрозалежною?*
8. *Що задають атрибути тегів і як вони записуються?*
9. *Описати структуру HTML-документа.*
10. *Перерахувати основні атрибути тегу body.*
11. *Яким тегом описують заголовки?*
12. *Який тег використовується для установки шрифту?*
13. *Які види списків дозволяє відображати HTML?*
14. *Як задаються різні види списків в HTML?*

## **Тема 10**

### **Мова гіпертекстової розмітки HTML. Створення таблиць. Робота з графічними об'єктами. Створення гіперпосилань**

Зображення в Інтернет бувають різних форматів. Найпоширенішими серед них є **GIF, JPG**.

Формат **GIF** дозволяє створювати анімовані картинки. Цей формат є стислим, і допустима кількість кольорів в ньому 256. Стиснення відбувається без втрат, тобто зображення записується без яких-небудь перетворень.

Формат **JPG** також є стислим, і застосовується для зображень, які вимагають великої кількості кольорів – до 16 млн. Стиснення відбувається після оптимізації і тому відбуваються втрати в якості. Розмір файлу сильно залежить від необхідної якості зображення, яке може мати 7 рівнів якості (від дуже поганого до якнайкращого).

Все більшого поширення при розробці веб-сторінок набуває формат **PNG**. Він дозволяє створювати зображення в 16 млн. кольорів без втрати в якості. Але через це займає набагато більше місця, ніж **JPG**.

**<img>** – вставляє в документ зображення. Атрибути:

**src** – вказується джерело зображення. Як правило – це шлях відносно поточного документа, але можна використовувати замість шляху його URL в Інтернет. Закриваючий тег не використовується;

**alt** – задає текст, який буде відображатися замість картинки, якщо вона не завантажилася або не відобразилася. Цей текст також з'являється в спливаючій підказці. Як правило в цьому тексті вказується короткий опис зображення або посилання, іноді міститься назва файлу з його розміром;

**lowsrc** – джерело зображення низької якості або розміру, для швидкого попереднього завантаження;

**align** – задає вирівнювання зображення відносно поточного рядка. Значення параметра:

**top** – вирівнює верхній край зображення по верхньому краю поточного рядка;

**middle** – вирівнює центр зображення по базовій лінії поточного рядка;

**bottom** – установка за замовчуванням. Вирівнює нижній край зображення по базовій лінії поточного рядка;

**left (right)** – вирівнює зображення по лівому (правому) полю;

**width i height** – горизонтальний і вертикальний розмір картинки;



**border** – задає товщину рамки, якою буде обрамлене зображення. Якщо зображення не є посиланням, то воно за замовчуванням не оточується рамкою. Якщо ж це зображення-посилання, то товщина рамки буде рівна 2;

**vspace** – ширина простору зверху і знизу від зображення в пікселях. Як правило за замовчуванням дорівнює 0.

Зауваження: В назвах файлів зображень доцільно використовувати тільки прописні (маленькі) букви алфавіту. Це пов'язано з тим, що на комп'ютерах, де використовуються Unix-системи, регістр символів має значення.

Відображення таблиці на веб-сторінці задається тегом **<table>**. Закриваючий тег обов'язковий. Атрибути:

**align** – задає вирівнювання таблиці. За замовчуванням таблиця вирівнюється по лівому краю;

**width** – ширина таблиці. Як правило браузер підбирає розмір таблиці автоматично. Для задання конкретного розміру його потрібно вказати в пікселях або відсотках від вільного простору;

**border** – ширина межі таблиці в пікселях. За замовчуванням має дорівнює 0 (межі таблиці не відображаються);

**cellspacing** – ширина простору між рамками (кожну комірку браузер обводить своєю власною рамкою);

**cellpadding** – ширина простору між рамкою елемента таблиці і її вмістом;

**bgcolor** – колір фону таблиці;

**background** – зображення, яке буде відображатися у вигляді фону таблиці.

**<caption> ... </caption>** – заголовок таблиці, який буде відображатися над таблицею. Даний тег записується відразу після **<table>**.

Далі починається сама таблиця:

**<tr>** – створення нового ряду комірок в таблиці. Закриваючий тег не обов'язковий;

**<td>** або **<th>** (**<th>=<td><b>...</b>**) – створення нової комірки (стовпця). Кількість комірок (тобто елементів **<td>** і **<th>**) повинна бути однаковою в кожному рядку таблиці, тобто усередині кожного тегу **<tr>**.

Атрибути тегу **<tr>**:

**align** – вирівнювання всередині всіх комірок таблиці;

**valign** – вирівнювання по вертикалі. Може приймати значення **top**, **middle**, **bottom**;

**bgcolor** і **background** – колір або зображення фону одного рядка таблиці;

**nowrap** – якщо вміст комірки потрібно відобразити в один рядок;

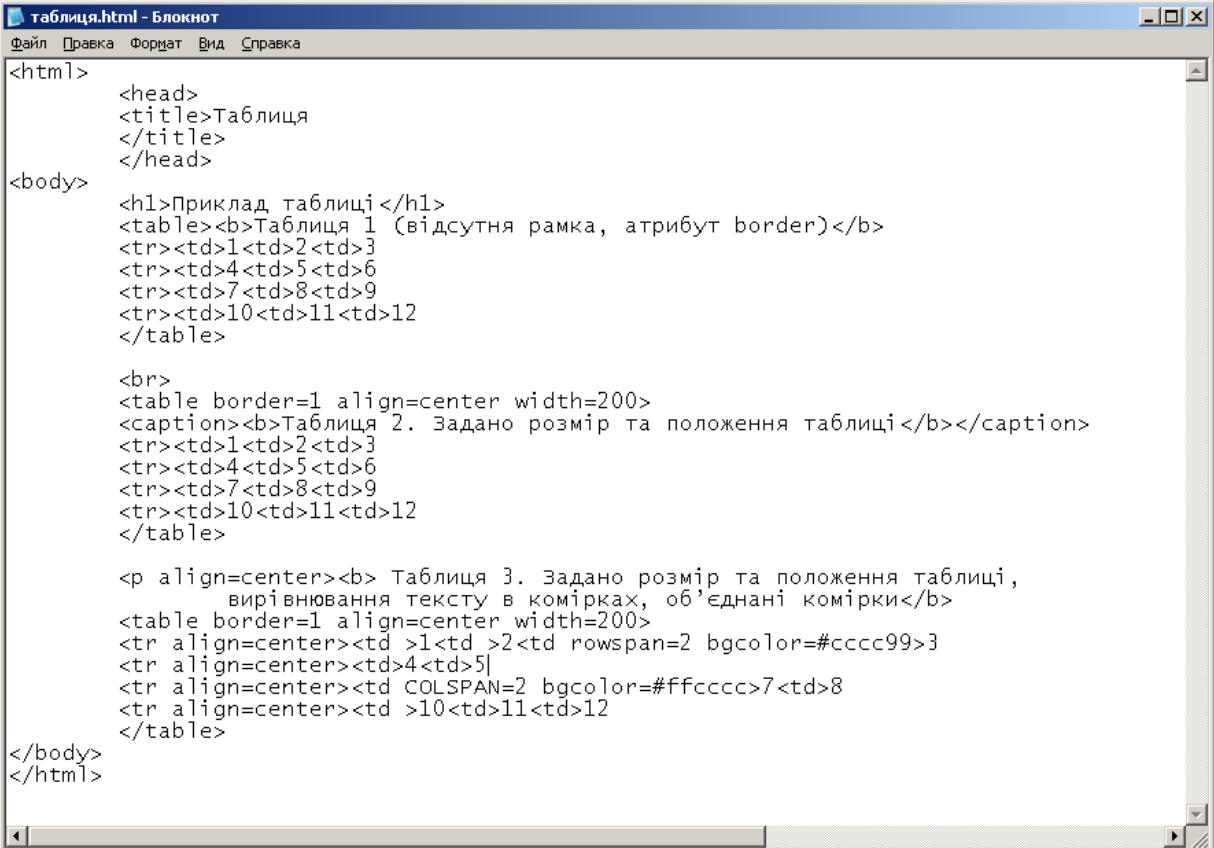
Атрибути тегу **<td>** є атрибути тегу **<tr>** і:

**rowspan** – об'єднує вказану кількість комірок в одну по вертикалі;

**colspan** – об'єднує комірки по горизонталі;

**width** і **height** – задають розміри комірки по горизонталі і вертикалі.

Наприклад, код, відображений на рис. 9, дозволить отримати три таблиці на веб-сторінці (рис. 10):



```
таблица.html - Блокнот
Файл  Правка  Формат  Вид  Справка
<html>
  <head>
    <title>Таблица
  </title>
</head>
<body>
  <h1>Приклад таблиці</h1>
  <table><b>Таблица 1 (відсутня рамка, атрибут border)</b>
  <tr><td>1<td>2<td>3
  <tr><td>4<td>5<td>6
  <tr><td>7<td>8<td>9
  <tr><td>10<td>11<td>12
  </table>

  <br>
  <table border=1 align=center width=200>
  <caption><b>Таблица 2. Задано розмір та положення таблиці</b></caption>
  <tr><td>1<td>2<td>3
  <tr><td>4<td>5<td>6
  <tr><td>7<td>8<td>9
  <tr><td>10<td>11<td>12
  </table>

  <p align=center><b> Таблица 3. Задано розмір та положення таблиці,
    вирівнювання тексту в комірках, об'єднані комірки</b>
  <table border=1 align=center width=200>
  <tr align=center><td >1<td >2<td rowspan=2 bgcolor=#cccc99>3
  <tr align=center><td>4<td>5|
  <tr align=center><td colspan=2 bgcolor=#ffcccc>7<td>8
  <tr align=center><td >10<td>11<td>12
  </table>
</body>
</html>
```

Рисунок 9

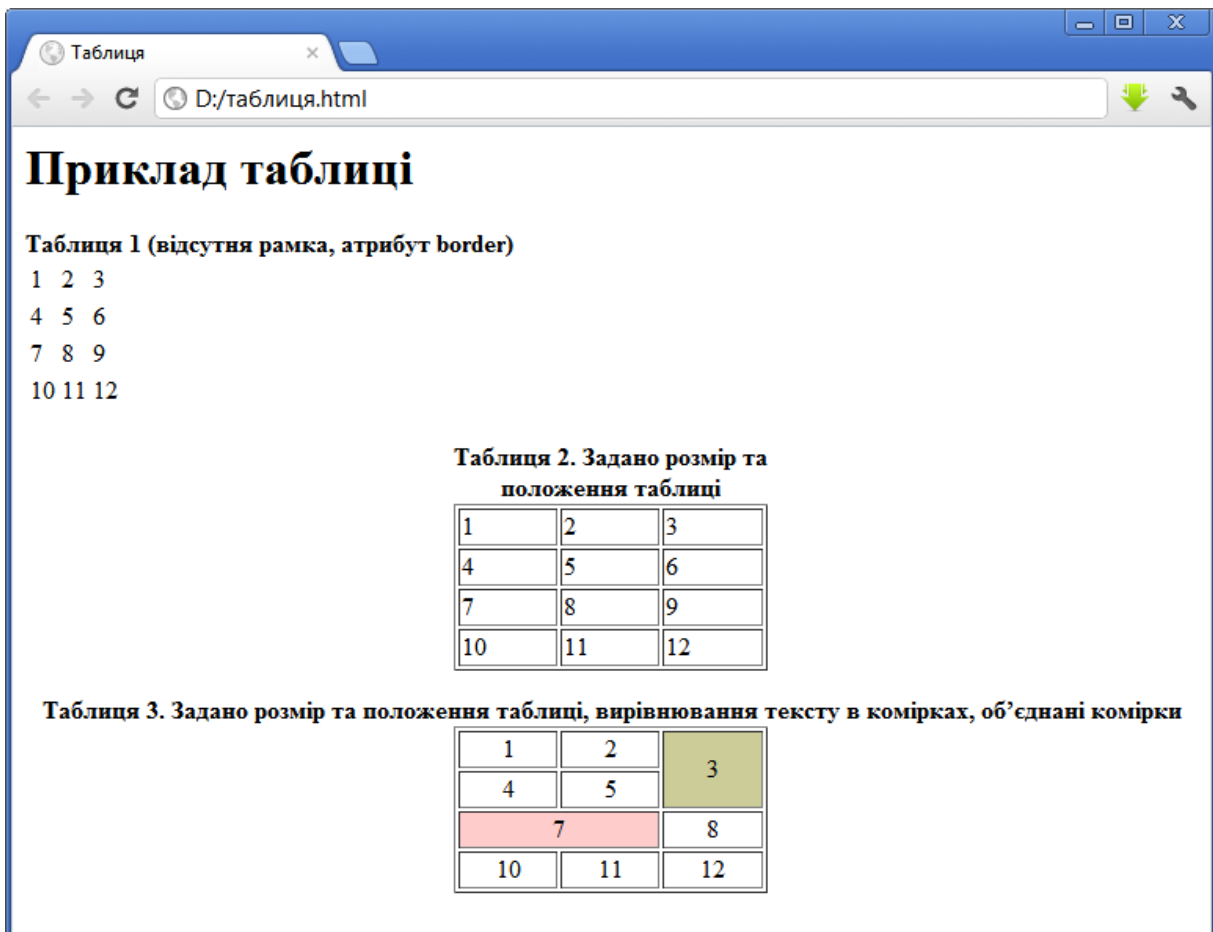


Рисунок 10

Для зв'язування сторінок (створення гіперпосилань) використовується тег **<a>**. Закриваючий тег обов'язковий. Атрибути:

**href** – задає посилання на інший документ. Сюди записується відносний шлях або конкретна URL;

**name** – задає назву для закладки;

**target** – задає ім'я фрейму, в якому буде відкриватися документ.

Для створення посилання на інший сайт або ресурс тег приймає вигляд:

`<a href=http://www.microsoft.com/> Посилання на ресурс в Інтернет</a>`

Закладки використовуються для створення змісту на головній сторінці сайту, для швидкого переходу на потрібний рядок (заголовок) в межах однієї сторінки. Закладка створюється з допомогою тегу **<a>** з атрибутом **name**. Наприклад:

`<a name=mark1>`

Замість mark1 можна використовувати іншу назву. Для використання закладок у посиланні вказується назва закладки після символу #. Якщо закладка розташовується в іншому файлі, використовується наступне посилання:

```
<a href=exam2.html#mark3>Закладка в другому документі</a>
```

В якості посилань можна використовувати зображення. Для цього всередині тегів <a>...</a> вказується тег <img src=...>. Наприклад:

```
<a href=http://www.rambler.ru/></a>
```

У цьому випадку зображення буде обведене рамкою шириною 2 пікселя.

### ***Запитання для самоконтролю***

- 1. Який тег використовується для вставки малюнку в веб-сторінку?*
- 2. Який атрибут є обов'язковим для тегу вставки малюнка на веб-сторінку?*
- 3. Назвати основні атрибути тегу img.*
- 4. Який тег призначений для вставки таблиці на веб-сторінку?*
- 5. Який тег призначений для вставки рядків і стовпців таблиці?*
- 6. Якими атрибутами задається ширина простору між рамкою комірки і текстом комірки?*
- 7. Якими атрибутами задається ширина простору між рамками?*
- 8. Перерахувати атрибути тегу задання рядка.*
- 9. Перерахувати атрибути тегу задання стовпця.*
- 10. Як задається заголовок таблиці?*

## **Тема 11**

### **Мова гіпертекстової розмітки HTML. Фрейми. Карта сайту. Форми**

Фрейми призначені для поділу робочого простору вікна браузера на частини (комірки) різних розмірів. У кожен таку частину потім можна буде завантажити окрему сторінку. Фрейми забезпечують зручну навігацію по сайту.

Для створення фреймів створюється окремий файл, який при завантаженні розбиває вікно і в якому міститься інформація про вміст кожної комірки.

При створенні окремого файлу замість тегу **<body>** задається тег **<frameset>**. Закриваючий тег обов'язковий. Атрибути:

**rows** – розбиває вікно на горизонтальні області;

**cols** – розбиває вікно на вертикальні області;

**frameborder** – ознака окантовки фрейму. Має значення **yes** або **no**;

**border** – задає ширину окантовки фрейму в пікселях;

**bordercolor** – колір окантовки фрейму.

Наприклад:

```
<frameset cols=100, 10%, *>
...
</frameset>
```

Розбити вікно можна на велику кількість областей. Після символу "=" вказуються всі розміри цих областей через кому. Розмір області може задаватися трьома способами: безпосередній розмір у пікселях, розмір у відсотках від розміру вікна й довільний розмір. В прикладі вікно розділиться на 3 вертикальні частини: перша шириною 100 пікселів, друга – 10% від розміру вікна, а розмір останньої – все, що залишається до повного розміру вікна.

**<frame>** – призначений для того, щоб вказати сторінки, які будуть завантажуватися у виділені області. Закриваючого тегу немає. Записується всередині **<frameset> ... </frameset>**. Атрибути:

**name** – вказується ім'я області (закладки), за яким потім можна буде до неї звертатися. Це ім'я повинне бути унікальним і не повторюватися;

**src** – ім'я файлу або URL документа, що буде завантажений в цю область;

**marginwidth/marginheight** – вертикальний/горизонтальний розмір межі фрейму в пікселях;

**frameborder** – ознака відображення межі фрейму. Має значення **yes** або **no**;

**scrolling** – відповідає за наявність полос прокрутки. Має значення **yes**, **no** або **auto**. Якщо задається “**yes**” скролінг буде присутній завжди, якщо “**auto**” – скролінг буде з’являтися при необхідності, якщо “**no**” – не з’являється ніколи;

**noresize** – забороняє зміну розміру фрейму. За замовчуванням розмір кожної області можна змінювати, перетягуючи межі з допомогою миші;

**bordercolor** – колір окантовки фрейму.

Щоб завантажити сторінку в певному фреймі, треба при описі посилання скористатися атрибутом **target**, у якому вказати ім’я необхідного фрейму.

Значення атрибуту **target**:

**\_top** – завантаження документа в основне вікно;

**\_parent** – завантаження документа в “батьківське” вікно по відношенню до поточного;

**\_self** – завантаження сторінки відбудеться в цьому ж фреймі, встановлюється за замовчуванням;

**\_blank** – завантаження сторінки в новому вікні.

Наприклад:

```
<a href="left.html" target="left1">Посилання завантажиться у фреймі з ім'ям left1 !!!</a>
```

На зміну фреймам частіше використовують блочний елемент **<div>**. Елемент **<div>** призначений для виділення фрагмента документа з метою зміни виду вмісту. Як правило, вигляд блоку налаштовується за допомогою стилів. Вміст тега **<div>** завжди починається з нового рядка.

Карти-зображення дозволяють прив’язувати посилання до різних областей одного зображення. Карта на вигляд нічим не відрізняється від звичайного зображення, але при цьому воно може бути розбите на невидимі зони різної форми, де кожна з областей служить посиланням.

Тег **<map>** служить контейнером для елементів **<area>**, які визначають активні області для карт-зображень. Такі області встановлюють невидимі зони на зображенні, що є посиланнями на HTML-документи. Мета використання тега **<map>** полягає в зв’язуванні тега **<img>** з клієнтською картою-зображенням.

Цей зв'язок визначається застосуванням єдиного ідентифікатора як в тегу **<img>**, що задається атрибутом **usemap**, так і в тегу **<map>**, що встановлюється атрибутом **name**.

Кожен елемент **<area>** визначає активні області зображення, які є посиланнями. Малюнок з прив'язаними до нього активними областями називається в сукупності картою-зображенням. Тег **<area>** задає форму області, її розміри, встановлює адресу документа, на який слід зробити посилання, а також ім'я вікна або фрейму, куди браузер буде завантажувати документ. Цей тег завжди розташовується в контейнері **<map>**, який пов'язує координати областей із зображенням. Кілька областей можуть перекривати одна одну, зверху буде та, яка в коді HTML розташовується вище.

Атрибути тегу **<area>**:

**shape** – визначає форму активної області. Форма може бути задана у вигляді кола (**circle**), прямокутника (**rect**), полігону (**poly**);

**alt** – додає альтернативний текст для кожної області. Служить лише коментарем для посилання, оскільки на екран не виводиться;

**coords** – визначає координати активної області. Відлік координат ведеться від лівого верхнього кута зображення, якому відповідає значення (0, 0). Перше число є координатою по горизонталі, друге – по вертикалі. Список координат залежить від форми області:

- для кола задаються три числа – координати центру кола і радіус.

Наприклад: `<area shape = "circle" coords = "230, 340, 100" href = "circle.html">`;

- для прямокутника – координати лівого верхнього і правого нижнього кута. Наприклад: `<area shape = "rect" coords = "24, 18, 210, 56" href = "rect.html">`;

- для полігону задаються координати його вершин;

**href** – визначає адресу посилання для області. Правила запису такі ж, як і для тегу **<a>**.

**Форми.** Тег **<form>** є контейнером для елементів управління і дає можливість вводити інформацію і відправляти її на сервер. Закриваючий тег обов'язковий. Атрибутами є **name**, **action**, **method**, **enctype**, **target**. Вони використовуються при відправленні даних форми для їх обробки скриптами.

Теги форми:

**<textarea>** – використовується для того, щоб дозволити користувачу вводити більше одного рядка інформації (вільний текст). Закриваючий тег обов'язковий. Якщо потрібно, щоб по замовчуванню в полі виводився якийсь текст, його необхідно вставити всередину тегів **<textarea>** і **</textarea>**.

Атрибути:

**name** – ім'я поля введення;

**rows (cols)** – висота (ширина) поля введення в символах;

**wrap** – визначає спосіб переносу слів в даній формі. Можливі значення:

**off** – перенесення слів не відбувається (значення за замовчуванням);

**virtual** – перенесення слів тільки відображається, на сервер поступає один неподільний рядок;

**physical** – перенесення слів буде відбуватися у всіх місцях переносу.

**<input>** – призначений для створення елементів управління і завжди використовується разом з атрибутом **type**, який визначає тип елемента управління. Закриваючий тег не обов'язковий. Атрибути:

**title** – впливаюча підказка;

**value** – задає текст за замовчуванням для **text**, **password**, **button**, **reset**, **submit**;

**size** – встановлює кількість видимих символів в **text**;

**name** – ім'я елемента. Обов'язковий атрибут, який використовується як унікальний ідентифікатор елемента;

**src** – URL, яка вказує на малюнок (використовується разом з атрибутом **image**);

**maxlength** – встановлює максимально допустиме число символів, які можна ввести в **text**;



**checked** (false або true) – приймає початковий стан для checkbox і radio (за замовчуванням false);

**type** – описує тип елемента. Може приймати наступні значення:

**text** – створює рядок для введення тексту;

**password** – створює рядок для введення тексту, при цьому відображає символи, які вводяться у вигляді зірочок;

**checkbox** – створює перемикач "прапорець";

**radio** – створює перемикач, який дозволяє вибрати одне значення з кількох альтернативних;

**file** – створює елемент для вибору локальних файлів. Наприклад, рядок для імені файла і кнопку "Обзор", при натисканні на яку відкривається діалог "Выбор файла";

**hidden** – створює невидимий для користувача елемент. Може використовуватись для відправки додаткової службової інформації (наприклад, пароль);

**image** – створює елемент форми у вигляді кнопки-зображення;

**button** – створює стандартну кнопку;

**submit** – створює кнопку "Подача запроса".

**reset** – створює кнопку "Сброс". При натисканні всі поля форми приймуть значення, дані їм за замовчуванням.

### **Меню вибору в формах.**

Під меню вибору в формах розуміють такий елемент інтерфейсу, як **listbox**. Існує три типи тегів меню вибору для форм:

- **<select>** – користувач вибирає одне значення з фіксованого списку значень, які представлені тегами **option**. Даний вид представляється як випадаючий **listbox**. Закриваючий тег обов'язковий. За замовчуванням перший елемент відображається в рядку вибору;

- **<select single>** – те саме, що **select**, але на екрані користувач бачить одночасно три елементи вибору. Якщо їх більше, то додається автоматичний

вертикальний скролінг. Кількість елементів, які одночасно відображаються, визначає атрибут **size**;

- **<select multiple>** – дозволяє вибрати кілька елементів з **listbox**.

Кількість одночасно вибраних елементів визначається атрибутом **multiple**.

Атрибути:

**name** – задає ім'я списку;

**size** – встановлює висоту меню в рядках, якщо воно має можливість вибору декількох елементів;

**multiple** – вказує на можливість вибору декількох елементів меню;

Тег **<option>** – використовується тільки з елементом **<select>**; описує окремі пункти меню. Закриваючого тегу немає. Атрибути:

**selected** – визначає пункт списку, який буде вибраний першим при завантаженні документа;

**value** – задає даному пункту унікальне ім'я, яке буде використовуватись разом із іншими відомостями про вміст заповненої форми.

### **Перемикачі.**

Перемикачі вводяться в форму з допомогою тегу **<input>**, для якого потрібно вказати **type = radio**. Атрибут **name** однаковий для всіх перемикачів, а атрибут **value** – має різні значення.

Наприклад:

```
<input type="radio" name="sex" value="male" checked>чол<br>  
<input type="radio" name="sex" value="female">жін<br>
```

### **Перемикачі “прапорці”.**

Перемикачі “прапорці” вводяться в форму з допомогою тегу **<input>**, для якого потрібно вказати **type = checkbox**. Атрибут **name** для всіх перемикачів різний, а атрибут **value** – однаковий.

Якщо треба, щоб якийсь прапорець або перемикач був вибраний за замовчуванням, то до відповідного тегу додається атрибут **checked**.

### **Запитання для самоконтролю**

1. Яким тегом задається вставка гіперпосилання на веб-сторінці?

2. Який атрибут є обов'язковим для тегу вставки гіперпосилання?
3. Для чого використовуються закладки? Як створити закладку?
4. Який тег використовується для створення фрейму?
5. Перерахувати обов'язкові та необов'язкові атрибути фрейму.
6. Який тег служить для початку задання карти-зображення?
7. Перерахувати атрибути тегу <area>.
8. Який тег призначений для вставки форми на веб-сторінку?
9. Перерахувати атрибути тегу form та описати їх призначення.
10. Якими тегами описуються елементи форми в HTML?

## Тема 12

### Каскадні таблиці стилів CSS. Поняття XHTML

**CSS** – Cascading Style Sheets – технологія опису зовнішнього виду документа, який написаний мовою розмітки. Іншими словами: це набір параметрів форматування, що застосовується до елементів документа, щоб змінити їх зовнішній вигляд. Як правило, використовується як засіб оформлення веб-сторінок в форматі HTML і XHTML, а також може застосовуватися до інших XML-документів.

Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (який здійснюється за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (який тепер здійснюється за допомогою формальної мови CSS).

#### Типи стилів

Розрізняють декілька типів стилів, які можуть спільно застосовуватися до одного документа. Це стиль браузера, стиль автора і стиль користувача.

#### Переваги стилів:

- розмежування коду і оформлення. В ідеалі, веб-сторінка повинна містити тільки теги логічного форматування, а вид елементів задається через

стилі. При подібному розподілі робота над дизайном і версткою сайту може вестися паралельно;

- різне оформлення для різних пристроїв. За допомогою стилів можна визначити вид веб-сторінки для різних пристроїв виведення: монітора, принтера, смартфона, КПК і ін. Наприклад, на екрані монітора відображати сторінку в одному оформленні, а при її друці – в іншому. Ця можливість також дозволяє приховувати або показувати деякі елементи документа при відображенні на різних пристроях;

- розширені в порівнянні з HTML способи оформлення елементів. На відміну від HTML стилі мають набагато більше можливостей по оформленню елементів веб-сторінок. Простими засобами можна змінити колір фону елемента, додати рамку, встановити шрифт, визначити розміри, положення та ін.;

- прискорення завантаження сайту. При зберіганні стилів в окремому файлі, він кешується і при повторному зверненні до нього витягується з кеша браузера. За рахунок кешування і того, що стилі зберігаються в окремому файлі, зменшується код веб-сторінок і знижується час завантаження документів;

- єдине стильове оформлення великої кількості документів. Сайт – це не просто набір зв'язаних між собою документів, але і однакове розташування основних блоків, і їх вигляд. Застосування одноманітного оформлення заголовків, основного тексту і інших елементів створює наслідуваність між сторінками і полегшує користувачам роботу з сайтом і його сприйняття в цілому. Розробникам використання стилів істотно спрощує проектування дизайну;

- централізоване зберігання. Стилi, як правило, зберігаються в одному або декількох спеціальних файлах, посилання на які вказується у всіх документах сайту. Завдяки цьому зручно правити стиль в одному місці, при цьому оформлення елементів автоматично мiняється на всiх сторiнках, якi пов'язанi з вказаним файлом.

**Синтаксис CSS.** Стильові правила записуються в форматі, відмінному від HTML. Основним поняттям виступає селектор – це деяке ім'я стилю, до якого

додаються параметри форматування. Як селектор виступають теги, класи і ідентифікатори. Загальний спосіб запису має наступний вигляд:

**селектор {властивість1: значення; властивість2: значення; ...}**

Спочатку пишеться ім'я селектора, наприклад, TABLE, це означає, що всі стильові параметри будуть застосовуватись до тегу <table>, потім у фігурних дужках записується стильова властивість, а її значення вказується після двокрапки. Стильові властивості розділяються між собою крапкою з комою, в кінці цей символ можна опустити.

CSS не чутливий до регістра, перенесення рядків, пропусків і символів табуляції, тому форма запису залежить від бажання розробника.

### **Правила застосування стилів:**

- **форма запису** – для селектора допускається додавати кожен стильову властивість і його значення окремо:

```
td {background: olive}
td {color: white}
td {border: 1px solid black}
```

Проте такий запис не дуже зручний. Тому доцільно використовувати такий спосіб:

```
td {background: olive;
    color: white;
    border: 1px solid black;
}
```

Ця форма запису більш наочна і зручна у використанні;

- **має пріоритет значення, вказане в кодї нижче** – якщо для селектора спочатку задається властивість з одним значенням, а потім та ж властивість, але вже з іншим значенням, то застосовуватися буде те значення, яке в кодї встановлене нижче;

- **значення** – у кожній властивості може бути тільки відповідне його функції значення. Наприклад, для color, який встановлює колір тексту, як значення неприпустимо використовувати числа;

- **коментарі** – щоб позначити, що текст є коментарем, застосовують наступну конструкцію `/* ... */`. Коментарі можна додавати в будь-яке місце CSS-документа, а також писати текст коментаря в декілька рядків. Вкладені коментарі недопустимі. Коментарі звичайно застосовують з метою налагодження або з метою навчання, а при викладанні сайту в мережу їх знищують;

- **значення стильових властивостей** – стильові властивості можуть приймати значення таких типів: рядок, число, відсотки, розмір, колір, адреса або ключове слово.

### **Підключення таблиці стилів.**

Коли браузер зчитує таблицю стилів, він форматує документ згідно цієї таблиці. Існує три способи підключення таблиці стилів:

**1) підключення зовнішньої таблиці стилів (зв'язані стилі)** застосовується в ситуаціях, коли один стиль визначається для багатьох сторінок. Для підключення зовнішньої таблиці стилів призначений тег `<link>`, який розміщується в розділі заголовка:

```
<head>
<link rel="stylesheet" type="text/css" href="first.css">
</head>
```

Зовнішню таблицю стилів можна створити в довільному редакторі. Файл зовнішньої таблиці стилів не може містити ніяких тегів HTML. Наприклад:

```
H1 {color: #000080;
    font-size: 200%;
    font-family: Arial;
    text-align: center;}
P {padding-left: 20px;}
```

Розширення файла таблиці стилів – `.css`. Значення параметрів `rel` і `type` залишаються незмінними незалежно від коду.

**2) підключення внутрішньої таблиці стилів (глобальні стилі)** застосовується тоді, коли один документ має унікальний стиль. Для визначення внутрішніх стилів використовують тег `<style>`, який розміщується в розділі заголовка:

```

<head>
<style type="text/css">
    li {font-family: arial}
    p {font-size: 25%}
    h1 {margin-right: 10pt}
</style>
</head>

```

**3) вбудовані стилі (внутрішні стилі)** використовуються в тому випадку, коли необхідно застосувати стиль для одного елемента. Недолік: він змішує вміст документа з його представленням і втрачає багато переваг таблиць стилів:

```

<body style="background-color: blue" >
<h1 style="color: yellow" >Заголовок H1</h1>
</body>

```

Всі описані способи використання CSS можуть застосовуватися як окремо, так і сумісно один з одним. Але при цьому першим завжди застосовується внутрішній стиль, потім глобальний, а в останню чергу – зв'язаний.

**Класи.** Класи застосовують, коли необхідно визначити стиль для індивідуального елемента веб-сторінки або задати різні стилі для одного тега. При використанні спільно з тегамі синтаксис для класів буде наступний:

**тег.ім'я\_класу {властиві1: значення; властиві2: значення; ...}**

Всередині стилю спочатку пишеться потрібний тег, а потім, через крапку, призначене для користувача ім'я класу. Імена класів повинні починатися з латинського символу і можуть містити в собі символ дефіса (-) і підкреслення (\_). Щоб вказати в коді HTML, що тег використовується з певним класом, до тегу додається параметр `class="ім'я_класу"`.

Можна, також, використовувати класи і без вказівки тега. Синтаксис в цьому випадку буде наступний:

**.ім'я\_класу {властивість1: значення; властивість2: значення; ... }**

При такому записі, клас можна застосовувати до будь-кого тегу.

Класи зручно використовувати, коли потрібно застосувати стиль до різних елементів веб-сторінки: елементів таблиці, посилань, абзаців і ін.

### **Одночасне використання різних класів**

До будь-кого тегу одночасно можна додати декілька класів, перераховуючи їх в параметрі class через пробіл. В цьому випадку до елемента застосовується стиль, описаний в правилах для кожного класу. Оскільки при додаванні декількох класів вони можуть містити однакові стильові властивості, але з різними значеннями, то береться значення з класу, який описаний в кодї нижче.

**Ідентифікатори.** Ідентифікатор (ID-селектор) визначає унікальне ім'я елемента, яке використовується для зміни його стилю і звернення до нього через скрипти. Синтаксис:

**#ім'я\_ідентифікатора {властивість1: значення; властивість2: значення; ... }**

Ім'я ідентифікатора повинне починатися з латинського символу і може містити в собі символ дефіса (-) і підкреслення (\_). На відміну від класів ідентифікатори повинні бути унікальними, тобто зустрічатися в кодї документа тільки один раз.

Звернення до ідентифікатора відбувається аналогічно класам, але як ключове слово у тега використовується параметр **id**, значенням якого виступає ім'я ідентифікатора. Символ решітки при цьому вже не вказується.

Як і при використанні класів, ідентифікатори можна застосовувати до конкретного тегу. Синтаксис при цьому буде наступний.

**тег#ім'я\_ідентифікатора {властивість1: знач; властивість2: знач; ... }**

XHTML (Extensible Hypertext Markup Language, розширювана мова розмітки гіпертексту) – сімейство мов розмітки веб-сторінок на основі XML, що повторюють і розширюють можливості HTML 4, призначена для заміни HTML і вважається його більш суворою версією.

Правила XHTML:



- всі теги і їх атрибути повинні бути набрані в нижньому регістрі (малими символами);
- значення будь-яких атрибутів необхідно брати в лапки;
- потрібно закривати всі теги, навіть такі, які не мають закриваючого тегу;
- повинна виконуватися правильна вкладеність тегів;
- не можна використовувати скорочені атрибути тегів;
- замість атрибуту name слід вказувати id;
- слід визначати DTD (document type definition – опис типу документа) за допомогою елемента `<!DOCTYPE>`.

### ***Запитання для самоконтролю***

1. *Що таке CSS?*
2. *Для чого призначені CSS?*
3. *Що таке селектор?*
4. *Описати загальний синтаксис задання селектору.*
5. *Які типи стилів існують?*
6. *Перерахувати переваги використання CSS.*
7. *Чи чутливі CSS до регістру, символів абзаців, пробілів та табуляції?*
8. *Якими способами можна підключати CSS до веб-сторінки?*
9. *Для чого призначений кожний із способів?*
10. *Як відбувається підключення CSS в кожному із випадків?*
11. *Для чого використовуються класи в CSS? Синтаксис написання класу.*
12. *Для чого використовуються ідентифікатори в CSS? Синтаксис написання ідентифікатора.*
13. *Що таке XHTML?*
14. *Чим відрізняється XHTML від HTML?*

## Лабораторна робота № 1

**Тема:** Сервіс WWW. Програми для роботи з WWW: браузери, робота в різних браузерах

**Мета:** Розглянути принципи роботи браузерів, набути навичок роботи із гіпертекстовими документами.

**Теоретичні відомості:** теми 4-5.

### Завдання

1. Завантажити браузери Internet Explorer, Mozilla Firefox, Opera та інші (за умови наявності на комп'ютері). Порівняти інтерфейс.

2. У завантажених браузерах завантажити веб-сторінки. Здійснити навігацію, використовуючи засоби різних браузерів (перехід за посиланням, перехід на попередню/наступну сторінку, перехід на домашню сторінку).

3. Розглянути засоби браузерів, призначені для зберігання вибраних посилань, для зберігання відвіданих сторінок протягом певного проміжку часу. Зберегти 2-3 посилання, зберегти 2-3 посилання в окремих розділах (папках). Переглянути, які сайти були завантажені на даному комп'ютері 3 дні тому.

4. Розглянути основні налаштування браузера: налаштування вигляду вікна браузера, зменшення/збільшення розміру шрифту веб-сторінки, зміна кодування, встановлення домашньої сторінки, вибір кількості днів, протягом яких зберігаються відвідані сайти.

5. Розглянути різні способи збереження веб-сторінок.

6. Порівняти роботу в наявних браузерах. Визначити схожі та відмінні засоби. Відзначити характерні особливості. Звіт про виконані завдання оформити у вигляді наступної таблиці:

Браузер	Сторінки, збережені 3 дні тому	Переваги	Недоліки	Особливості

## Лабораторна робота № 2

**Тема: Електронна пошта: створення власної електронної скриньки та основні прийоми роботи**

**Мета:** Навчитись користуватись сервісом електронної пошти, використовуючи засоби браузера.

**Теоретичні відомості:** теми 5, 7.

### Завдання

1. Знайти сайти, які надають користувачу можливість безкоштовного створення та роботи з електронною скринькою.

2. На одному-двох сайтах (у випадку двох – обов'язково на gmail.com) створити електронну скриньку, заповнивши відповідну форму реєстрації. У випадку створення двох електронних скриньок, порівняти процес реєстрації.

3. Переглянути інтерфейс електронних скриньок. Визначити призначення папок електронної скриньки

4. Перейти в режим написання електронного листа. Порівняти інтерфейси. Визначити призначення полів для заповнення при написанні електронного листа. Виділити обов'язкові та необов'язкові поля.

5. Розглянути можливості набору форматowanego тексту в електронному листі.

6. Надіслати листи на вибрану адресу. Отримати електронні листи. Відповісти на отримані листи.

7. Порівняти роботу різних електронних скриньок. Визначити переваги, недоліки та особливості.

8. Звіт про виконані завдання оформити у вигляді наступної таблиці:

Поштовий сервер	Максимальний об'єм скриньки	Максимальний об'єм листа	"Термін придатності"	Обмеження на тип прикріплених файлів	Переваги	Недоліки	Особливості

Оформлений звіт надіслати у прикріпленому файлі на електронну скриньку (адреса повинна бути вказана викладачем).

### Лабораторна робота № 3

**Тема: Пошук інформації з допомогою пошукових каталогів та пошукових систем. Простий пошук.**

**Мета:** Навчитись користуватись пошуковими каталогами та використовувати простий пошук у інформаційно-пошукових системах.

**Теоретичні відомості:** теми 5-6.

#### Завдання

**Завдання 1.** Використовуючи пошукові каталоги, знайти:

1. Інформацію про вакансію секретаря у м. Луцьку.
2. Характеристики останньої моделі мобільного телефону Samsung.
3. Приблизну вартість цифрової фоторамки та місце їх продажу у місті Луцьку.

**Завдання 2.** Використовуючи 3 пошукові системи, знайти відповіді на наступні питання:

1. Дата народження вченого А.К.Кея.
2. Коли та де відбулись перші зарубіжні гастролі гурту Beatles?
3. Місце та дата народження математика М. І. Лобачевського.
4. Рік заснування та перша назва компанії "Nike".
5. Книги по Delphi російською мовою, які були надруковані пізніше 2010 року.
6. Визначення поняття "АСУ". Приклади використання АСУ в Україні.
7. Означення поняття "нейромережа".
8. Перший випуск Вісника Львівської політехніки.
9. Перерахувати конференції з інформаційних технологій, що відбудуться у першому півріччі поточного навчального року.
10. Скільки сантиметрів в 15 дюймах?

Звіт про виконання завдання подати у наступному вигляді:

Результати пошуку				
Ключова фраза	Посилання на джерело	Google	Yandex	Rambler

Зробити висновок щодо ефективності здійснення пошуку необхідної інформації використаними інформаційно-пошуковими системами.

#### Лабораторна робота № 4

**Тема: Пошукові системи: пошук інформації в Інтернет засобами складного пошуку. Мова пошукових запитів**

**Мета:** Навчитись використовувати мову запитів та форми інформаційно-пошукових систем для організації складного пошуку.

**Теоретичні відомості:** теми 5-6.

#### Завдання

**Завдання 1.** Використовуючи 2-3 пошукові системи, знайти відповіді на наступні питання:

1. В якому році і який льотчик першим перелетів через протоку Ла-Манш?

2. Хто пілотував і як називався літак, на якому вперше було здійснено безпосадковий кругосвітній переліт? В якому році був здійснений цей переліт?

3. З якого питання і хто з президентів США консультувався із Нобелівським лауреатом 1939 року з фізики?

4. Назвати справжнє прізвище письменника, який написав твір "Ріка Хронос". Перерахувати ще 4 його твори.

5. Знайти фотографію останньої випущеної автомашини Nissan Almera Classic. Вказати ціну автомобіля у офіційного дилера у регіоні.

6. Хто виконав пісню "Крылатые качели" у фільмі "Приключения Электроника"?

Звіт про виконання завдання подати у наступному вигляді:

Результати пошуку				
Ключова фраза	Посилання на джерело	Google	Yandex	Rambler

Зробити висновок щодо ефективності здійснення пошуку необхідної інформації використаними інформаційно-пошуковими системами.

**Завдання 2.** У довільній пошуковій системі сформулювати запит для пошуку інформації про електронні книги. Із пошуку слід виключити рекламу, посилання на Інтернет-магазини, реферати. Пошук слід зосередити на принципах функціонування таких пристроїв та їх характеристиках (завдання виконати за допомогою формулювання складних запитів).

Один і той же сформульований пошуковий запит використати у 2-3 інформаційно-пошукових системах. Звіт про виконання завдання повинен містити сформульований запит та висновок щодо ефективності пошуку у різних інформаційно-пошукових системах.

## **Лабораторна робота №5**

**Тема: Робота із сервісами Google**

**Мета:** Навчитись використовувати засоби сервісів Google.

**Теоретичні відомості:** тема 5, тема 8.

### **Завдання**

**Завдання 1.** Створити обліковий запис електронної пошти на Gmail. (Якщо у Вас уже є адреса електронної пошти Gmail, використати її для виконання наступних завдань).

**Завдання 2.** Створити акаунт на сервісі GoogleDocs. Розглянути роботу засобів, які входять до його складу: роботу з текстовими документами, електронними таблицями, презентаціями, створення графічних зображень. Для кожного із цих засобів створити по одному демонстраційному файлу-прикладу їх використання. Файл електронних таблиць зробити загальнодоступним, інші дозволити використовувати лише власнику облікового запису.

**Завдання 3.** Із використанням сервісу «Веб-альбоми Picasa» створити 2 власні альбоми (2-3 зображення у кожному). Один із них зробити

загальнодоступним, для іншого – надати можливість переглядати фото лише власнику облікового запису.

**Завдання 4.** У Google календар створити нагадування про такі події:

- дні народження 2-3 друзів, додати до події мінімум 2 гостей і дозволити їм переглядати подію та запрошувати інших людей до перегляду події;

- день народження одного члена вашої родини. Дозволити лише приватний перегляд цієї події. Додати нагадування через виринаюче вікно за один день до події; (*зауваження:* при виконанні цих завдань не обов'язково вказувати достовірну інформацію про дати народження друзів та членів родини);

- дату закінчення поточного навчального семестру. Дозволити переглядати подію усім. Додати нагадування за допомогою електронної пошти за тиждень до події.

**Завдання 5.** На сервісі YouTube відшукати освітній канал, тема якого стосується інформаційних технологій. Використовуючи обліковий запис Gmail, здійснити підписку на такий канал.

## Лабораторна робота № 6

**Тема:** Створення веб-сторінок засобами HTML

**Мета:** Навчитись розробляти структуру веб-сторінки та здійснювати форматування засобами мови HTML.

**Теоретичні відомості:** теми 9-10.

### Завдання

Засобами мови гіпертекстової розмітки HTML створити веб-сторінки за поданим зразком (рис. 1, рис. 2).

Параметрами для шрифтів та фону:

Сторінка 1. Шрифт: "book Antiqua", колір шрифту #006600, розмір – 4, фон – "sitka.gif".

Сторінка 2. Шрифт: "book Antiqua", колір #000099, розмір – 4, для коду програми шрифт "verdana", колір "black", колір фону – білий.

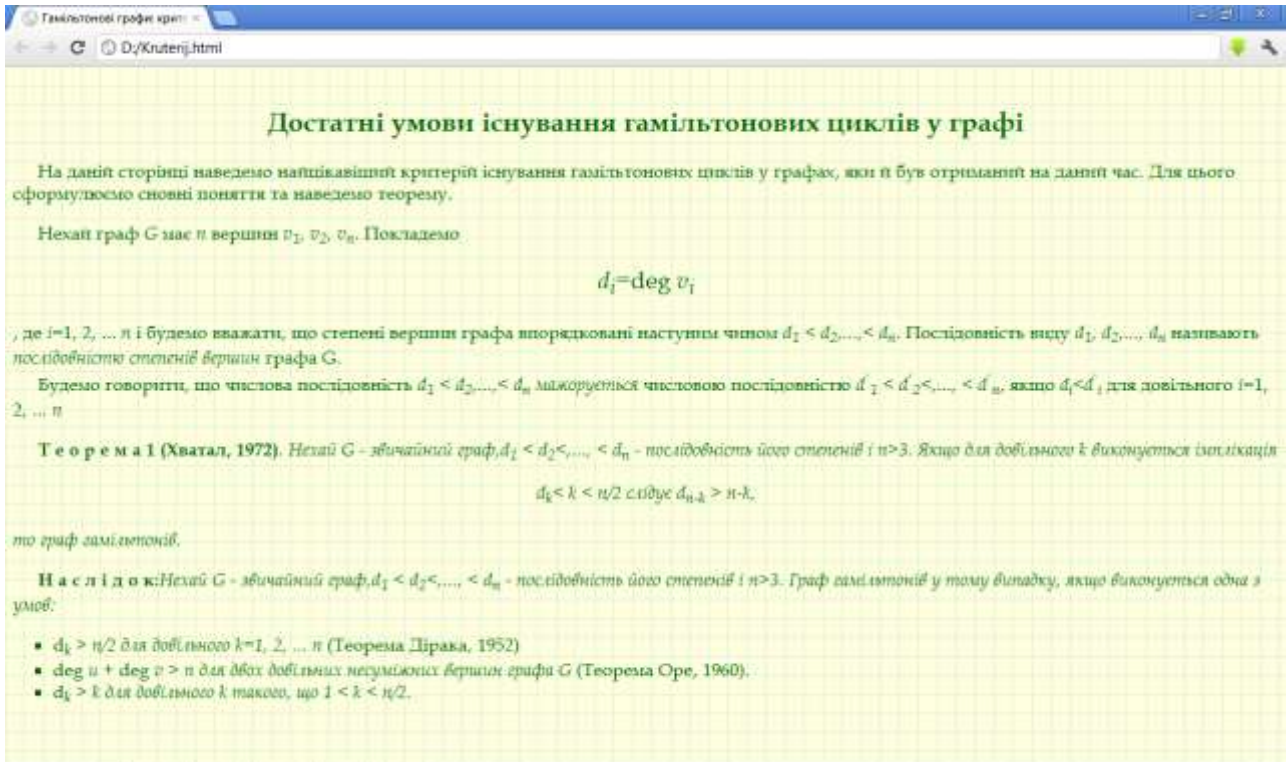


Рисунок 1



Рисунок 2



## Лабораторна робота № 7

**Тема: Вставка графічних зображень та таблиць на веб-сторінки**

**Мета:** Навчитись розміщати на веб-сторінках графічні зображення та оформляти інформацію у вигляді таблиць різної структури засобами HTML.

**Теоретичні відомості:** теми 9-11.

### Завдання

**Завдання 1:** Засобами мови гіпертекстової розмітки HTML створити веб-сторінку за поданим зразком (рис. 1, рис. 2, рис. 3).

Параметрами для шрифтів та фону: шрифт – Comic Sans MS, колір шрифту – #000099, розмір – 4, колір фону – білий.

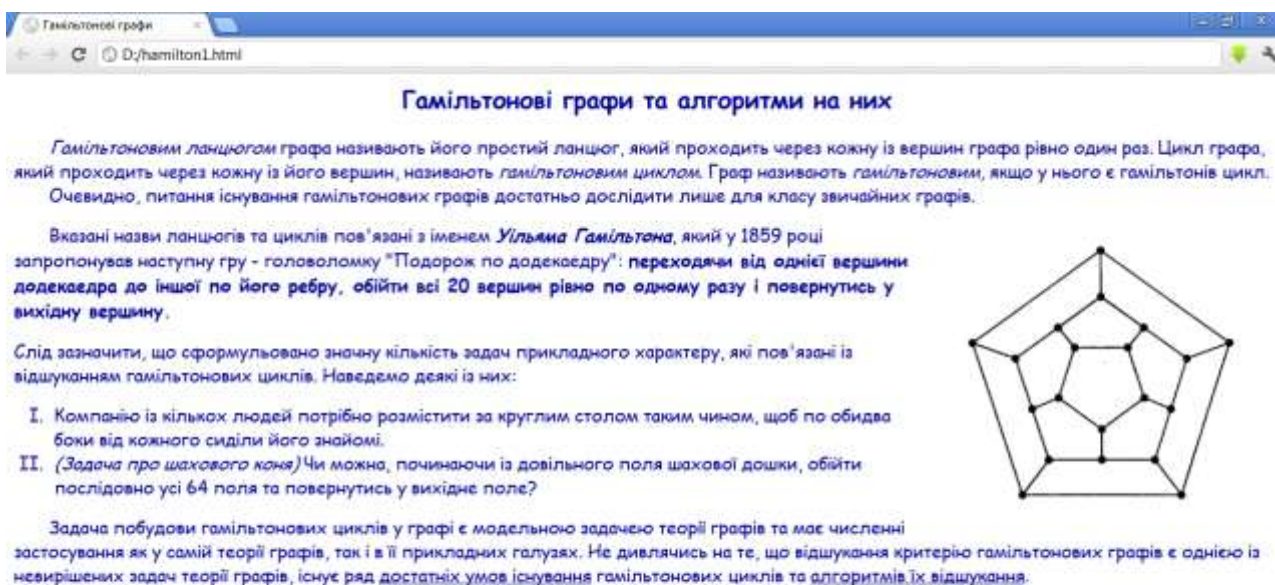


Рисунок 3

**Завдання 2:** Створити таблиці за наступним зразком:

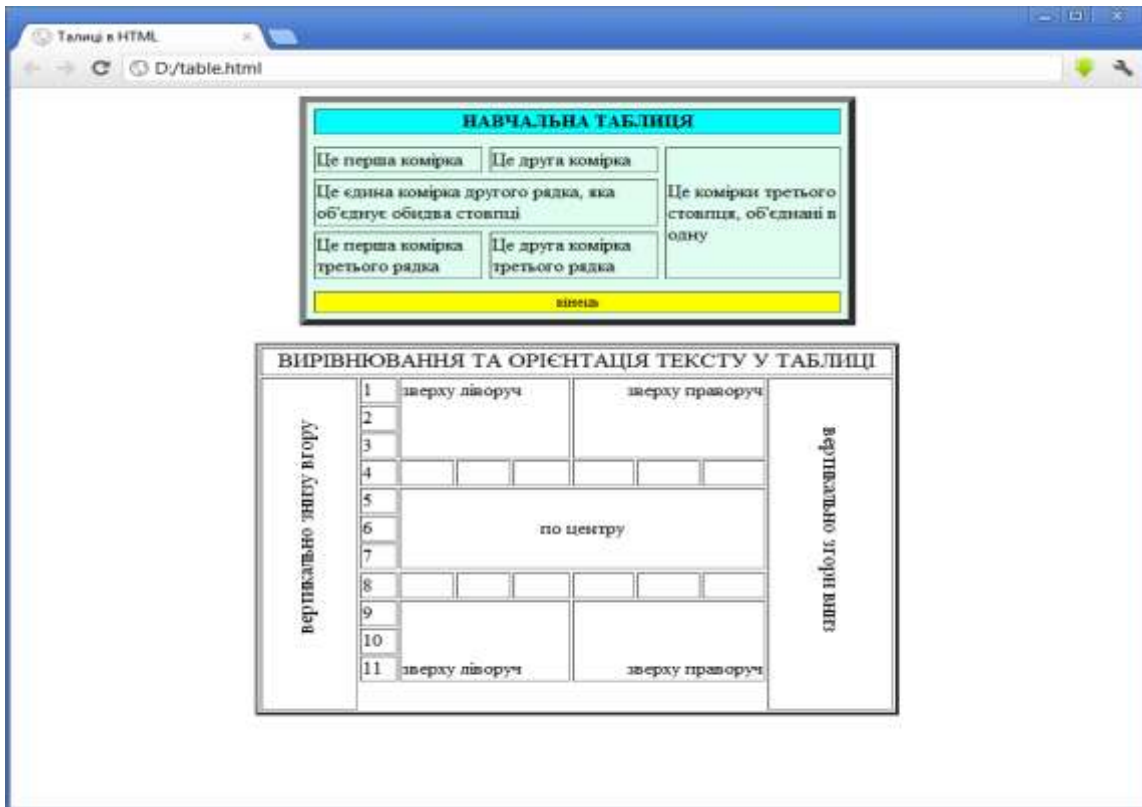


Рисунок 4

## Лабораторна робота № 8

### Тема: Створення гіперпосилань. Карти-зображення

**Мета:** Навчитись організовувати зв'язок між веб-сторінками за допомогою гіперпосилань, створювати карти-зображення.

**Теоретичні відомості:** теми 9-11.

### Завдання

**Завдання 1:** Створити гіперпосилання на сторінках, розроблених на попередньому занятті:

**Сторінка 1** "Гамільтонові граfi": текстовий фрагмент "ряд достатніх умов існування" – посилання на сторінку "Гамільтонові граfi: критерій", текстовий фрагмент "алгоритмів їх відшукання" – посилання на сторінку "Гамільтонові граfi: алгоритм". Колір посилання – той же, що і колір шрифту, колір перейденого посилання – зелений, колір активного посилання – рожевий. Сторінки повинні завантажуватись у тому ж вікні.

**Сторінка 2** "Гамільтонові графи: критерій": Теорема Оре, теорема Дірака – посилання на Інтернет-сторінки із формулюванням зазначених теорем (сторінки та їх адреси відшукати самостійно). Сторінки повинні завантажуватись у новому вікні.

**Завдання 2:** Створити сторінку за зразком (рис. 4). Використовуючи карти-зображення, створити посилання на наступні сторінки:

- напис "задача" – на сторінку 1 "Гамільтонові графи".
- напис "критерії" – на сторінку 1 "Гамільтонові графи: критерій".
- напис "алгоритм" – на сторінку 1 "Гамільтонові графи: алгоритм".
- напис "теорія графів" – на сторінку, розміщену у мережі Інтернет, що містить відомості про теорію графів, як один із розділів дискретної математики.

Сторінки повинні відкриватись у тому ж вікні.



Рисунок 5

## Лабораторна робота № 9

**Тема: Фреймова структура документа**

**Мета:** Навчитись створювати HTML-сторінки із фреймами, задавати області відкривання сторінок у сторінках із такою структурою.

**Теоретичні відомості:** теми 9-11.

### Завдання

З використанням фреймів створити html-документ наступної структури:

<b>40%</b>	<b>50%</b>
	<b>50%</b>

У лівій частині документу відобразити файл із двома посиланнями, перше з яких відкриватиме сторінку у верхньому вікні правої частини, а друге – у нижній частині того ж вікна.

Файл лівої частини вікна повинен містити 2 посилання: 1 – на он-лайн посібник із мови HTML, 2 – на он-лайн посібник із мови CSS.

## Лабораторна робота № 10

**Тема: Валідність веб-сторінки. Стандарт doctype. Валідація документів.**

**Використання тегу <meta>**

**Мета:** Навчитись використовувати валідатор для перевірки сторінок на валідність, використовувати стандарт doctype при розробці сторінок; правильно задавати атрибути тегу <meta> та використовувати його при розробці веб-сторінок.

**Теоретичні відомості:** теми 9-11.

### Завдання

**Завдання 1:** Перевірити розроблені на лабораторних заняттях № 6, 7, 8 сторінки за допомогою валідатора ([www.w3c.org](http://www.w3c.org)) та виправити існуючі помилки.

**Завдання 2:** Для розроблених сторінок на лабораторних заняттях № 6, 7, 8 описати тег <meta>, де вказати усі необхідні атрибути. Перевірити їх за допомогою валідатора.

## Лабораторна робота № 11

**Тема:** Створення форм користувача у HTML

**Мета:** Навчитись створювати форми з основними елементами управління для організації зворотного зв'язку із користувачем.

**Теоретичні відомості:** теми 9-11.

### Завдання

**Завдання 1:** Використовуючи засоби створення форм користувача мови HTML, створити сторінку за поданим нижче зразком (рис. 6):

Приклад використання фс x

D:/Forms.html

### Анкета користувача

Особисті дані	Захоплення
Ваше ім'я: <input type="text"/>	<input checked="" type="checkbox"/> Комп'ютери та Інтернет
Ваш пароль: <input type="password"/>	<input type="checkbox"/> Література та мистецтво
Ваше фото: <input type="button" value="Вибрати файл"/> Файл не вибрано	<input type="checkbox"/> Музика
	<input type="checkbox"/> Автомобілі
	<input type="checkbox"/> Спорт та активний відпочинок
Місце проживання	Коментарі
--Виберіть місто-- Київ Луцьк Львів Одеса	<input type="text"/>
--Виберіть країну--	<input checked="" type="radio"/> Я вперше на сторінці
	<input type="radio"/> Я тут не вперше
<input type="button" value="Надіслати"/> <input type="button" value="Очистити"/>	

Рисунок 6

Параметрами для шрифту та фону: Comic Sams MS; колір: шрифту – #003300; комірок із заголовками – #00cc33; комірок – #ccffcc; комірки із кнопками – #006633.

**Завдання 2:** Для розробленої сторінки описати тег <meta>, де вказати усі необхідні атрибути. Перевірити розроблену сторінку за допомогою валідатора.

## Лабораторна робота № 12

**Тема: Каскадні таблиці стилів: форматування символів з допомогою CSS**

**Мета:** Навчитись використовувати засоби каскадних таблиць стилів для форматування тексту HTML-сторінки.

**Теоретичні відомості:** тема 12.

### Завдання

**Завдання:** Використовуючи засоби CSS для форматування тексту, створити сторінку за поданим зразком (рис. 7).

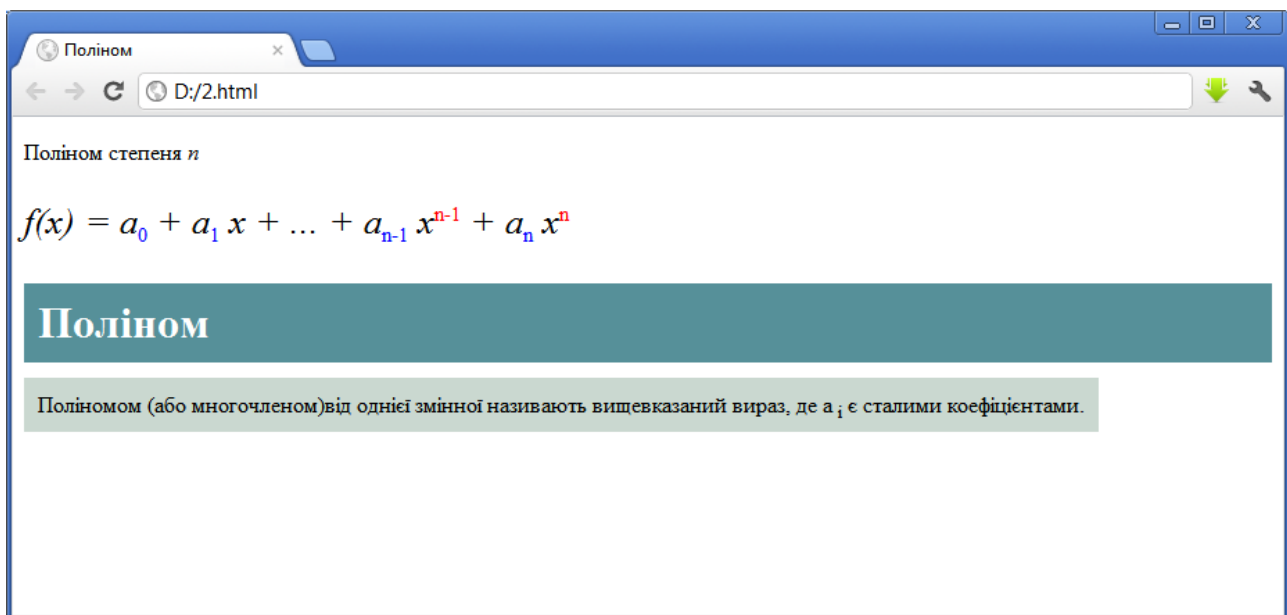


Рисунок 7

Параметрами для шрифту та фону: TimesNewRoman; колір першого блоку – #569099; розмір шрифту у першому блоці – h1; колір другого блоку – #cad8d04.

Стиль зберегти у зовнішньому файлі.

## Лабораторна робота № 13

### Тема: Каскадні таблиці стилів: використання класів

**Мета:** Навчитись використовувати класи для оголошення стилів форматування; розвивати вміння та навички форматування веб-сторінок засобами каскадних таблиць стилів.

**Теоретичні відомості:** тема 12.

### Завдання

Використовуючи засоби каскадних таблиць стилів, створити сторінку за поданим зразком (рис. 8, рис. 9).

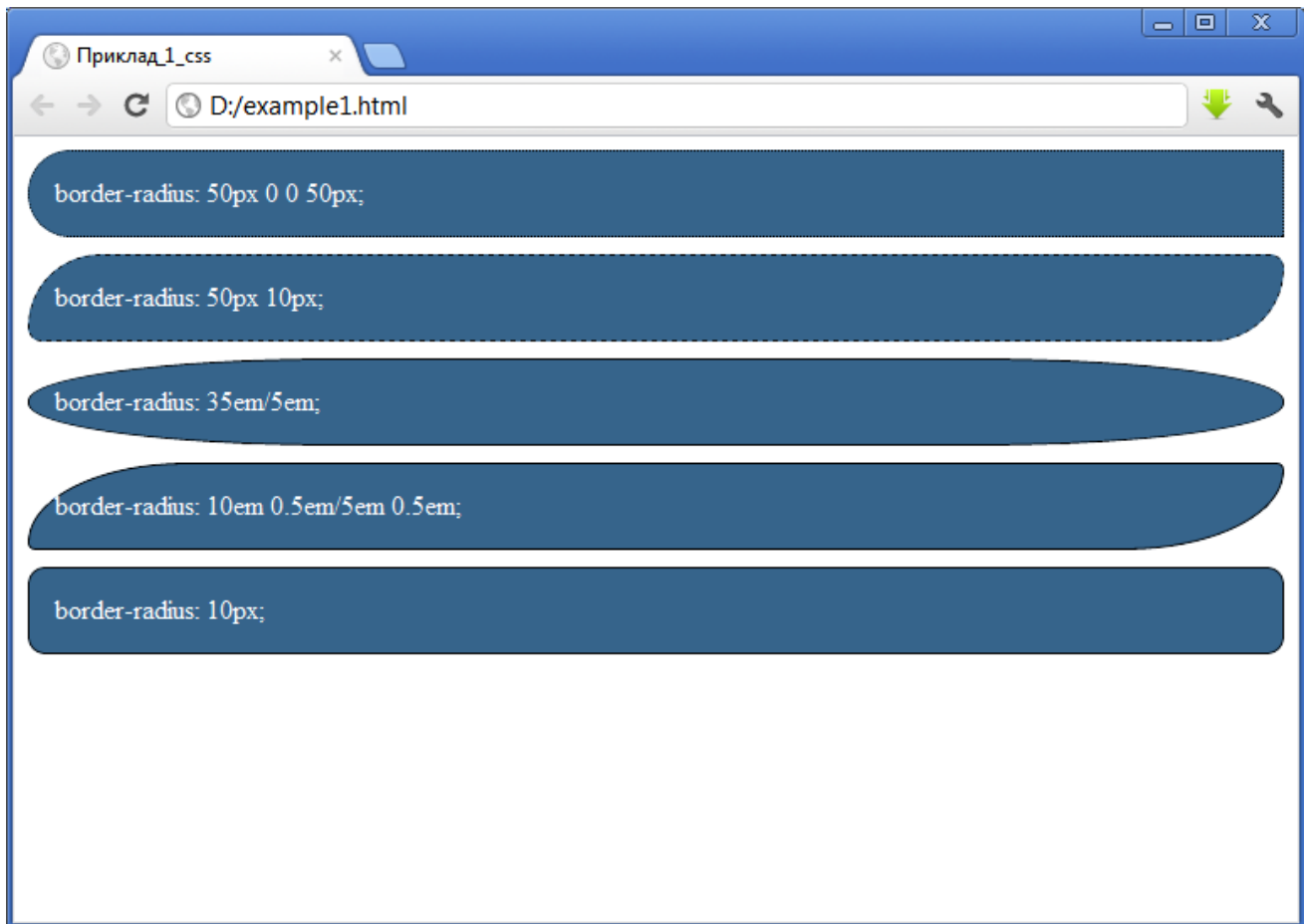


Рисунок 8

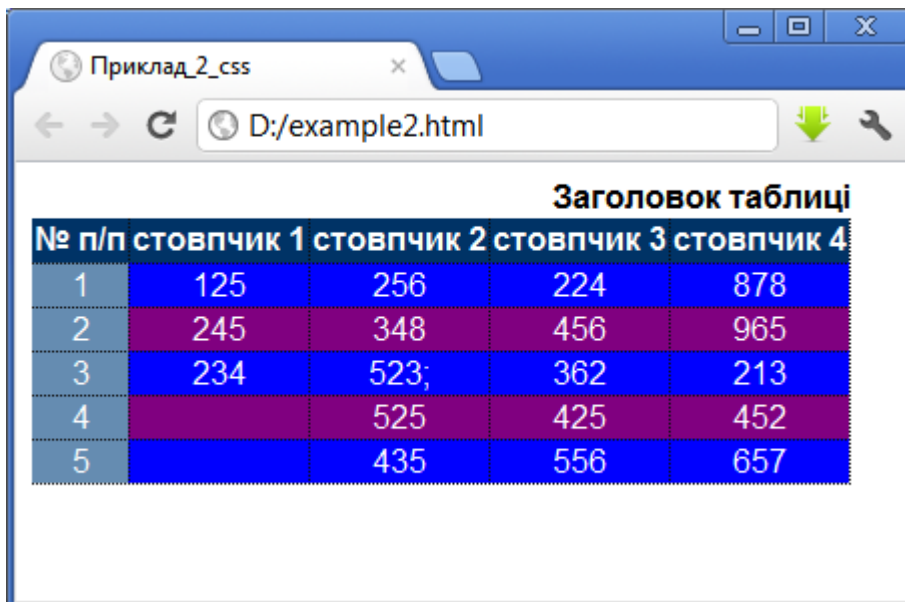


Рисунок 9

Параметрами для шрифту та фону (сторінка 1): шрифт – по замовчуванню, колір шрифту – #fff, колір фону для блоку – #36648b.

Параметрами для шрифту та таблиці (сторінка 2): шрифт – по замовчуванню, колір шрифту – #fff; колір заголовка таблиці – #036; колір першого стовпця – #658cb1; колір непарного рядка – #b0c47d; колір парного рядка – purple.

Стиль зберегти у зовнішньому файлі.

На етапі розробки (редагування) сторінки зовнішній файл, у якому описано стиль оформлення сторінки на рис. 10, матиме наступний вигляд:

```

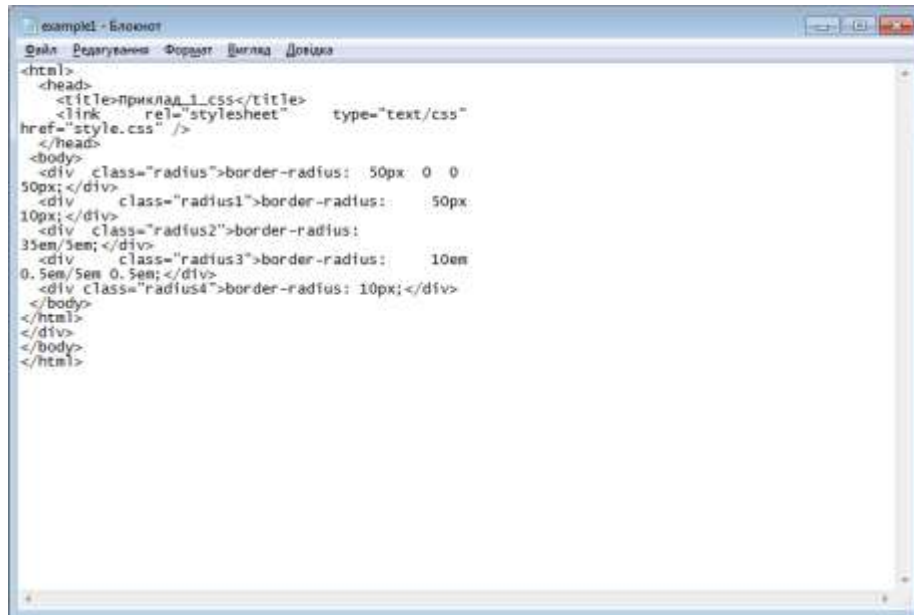
style - блокнот
Файл Редагування Формат Видгляд Довідка
.radius {
border-radius: 50px 0 0 50px;
background: #36648b;
border: 1px dotted black;
padding: 15px;
margin-bottom: 10px;
color: #fff;
}
.radius1 {
border-radius: 50px 10px;
background: #36648b;
border: 1px dashed black;
padding: 15px;
margin-bottom: 10px;
color: #fff; }
.radius2 {
border-radius: 35em/5em;
background: #36648b;
border: 1px solid black;
padding: 15px;
margin-bottom: 10px;
color: #fff;
}
.radius3 {
border-radius: 10em 0.5em/5em 0.5em;
background: #36648b;
border: 1px solid black;
padding: 15px;
margin-bottom: 10px;
color: #fff;
}
.radius4 {
border-radius: 10px;
background: #36648b;
border: 1px solid black;
padding: 15px;
}

```

Рисунок 10



Сама ж сторінка у вікні редактора виглядатиме наступним чином:



```
example1 - Блокнот
Файл  Редагування  Формат  Видгляд  Довідка
<html>
<head>
<title>Приклад_1_css</title>
<link rel="stylesheet" type="text/css"
href="style.css" />
</head>
<body>
<div class="radius">border-radius: 50px 0 0
50px;</div>
<div class="radius1">border-radius: 50px
10px;</div>
<div class="radius2">border-radius:
35em/5em;</div>
<div class="radius3">border-radius: 10em
0.5em/5em 0.5em;</div>
<div class="radius4">border-radius: 10px;</div>
</body>
</html>
</div>
</body>
</html>
```

Рисунок 11

## Лабораторна робота № 14

**Тема: Каскадні таблиці стилів: використання CSS для відображення веб-сторінки на різних пристроях**

**Мета:** Закріпити вміння та навички форматування веб-сторінок засобами каскадних таблиць стилів.

**Теоретичні відомості:** тема 12.

### Завдання

Використовуючи засоби каскадних таблиць стилів, створити сторінку за поданим зразком (рис. 12):

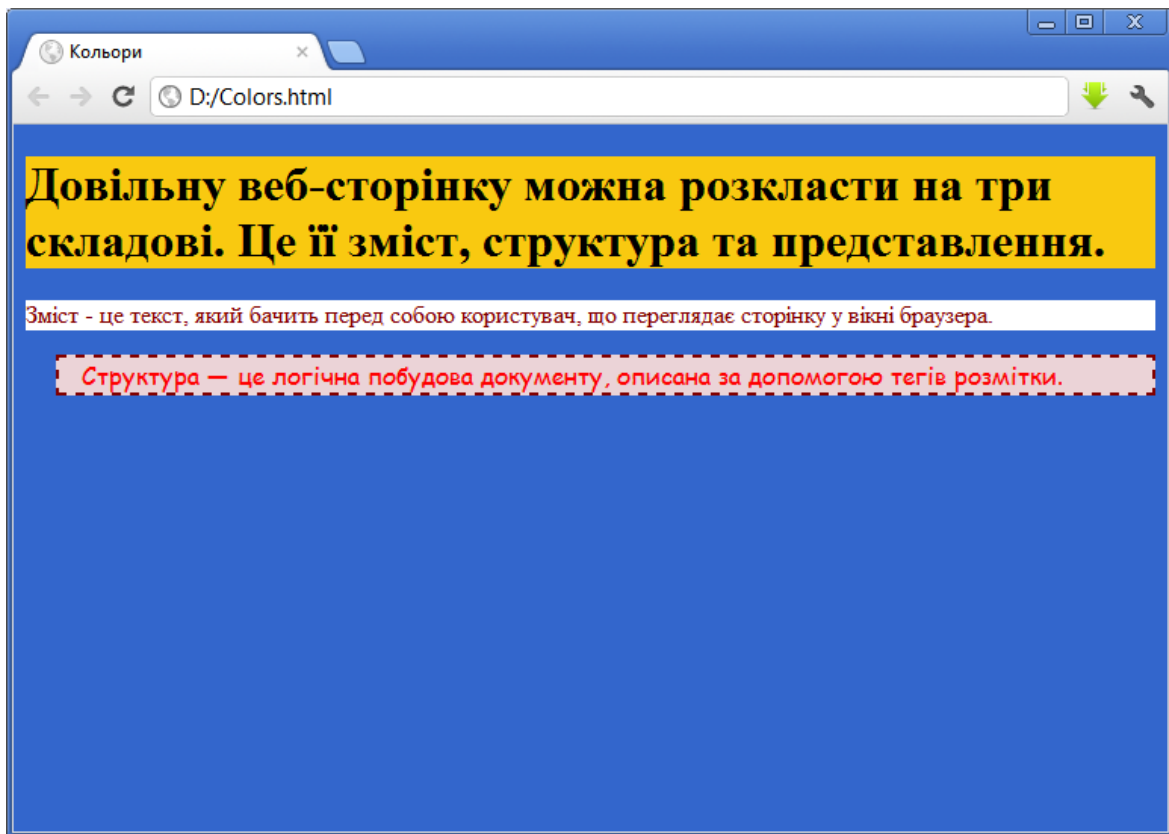


Рисунок 12

Параметрами для шрифту та фону (браузер): шрифт – по замовчуванню, колір шрифту блоку 1 – black; колір шрифту блоку 2 – maroon; колір шрифту блоку 3 – red; колір фону для блоку 1 – RGB(249, 201, 16); колір фону для блоку 2 – white; колір фону для блоку 3 – #ebd3d7.

Параметрами для шрифту та фону (друк): колір фону – білий, колір тексту – чорний, назва шрифту – Arial.

Стиль зберегти у зовнішньому файлі.

## Запитання та завдання для самостійної роботи

- 1) Перерахувати способи підключення до мережі Інтернет. Для підготовки відповіді використати джерела 1, 13-16 із рекомендованого списку.
- 2) Скориставшись джерелами 1, 8, 9, 15-17, з'ясувати, яка різниця між Інтернет та Web?
- 3) Перерахувати характеристики та принципи роботи DNS. Для підготовки відповіді на питання скористатись джерелами 1, 8, 9, 15-17 із рекомендованого списку.
- 4) Назвати служби Інтернет, які не описані у теоретичному матеріалі, та їх призначення. Для підготовки відповіді на питання слід скористатись джерелами 9, 15-17, 20 із рекомендованого списку.
- 5) Знайти та ознайомитися з роботою форумів, блогів, чатів. В чому їх різниця?
- 6) Знайти та ознайомитися з роботою інтернет-магазинів, інтернет-аукціонів.
- 7) Перерахувати відомі соціальні мережі та ознайомитися з їх роботою. Знайти приклади соціальних мереж для науковців чи людей за іншими інтересами.
- 8) Яке програмне забезпечення використовується для служби миттєвих повідомлень?
- 9) Чим відрізняється завантаження файла з Інтернет через FTP і файлообмінну мережу? Завантажити файл різними способами.
- 10) Назвати програмні засоби (продукти), які використовуються для відозв'язку у мережі Інтернет?
- 11) Використавши джерела 2, 3, 5, 7, 12 із рекомендованого списку, з'ясувати: як розмістити розроблену веб-сторінку у мережі? Як здійснити індексування цієї сторінки, щоб вона була "помічена" кроулером?
- 12) Навести приклади HTML-редакторів (використати джерела 2, 5-7, 12). Ознайомитись із принципами роботи таких програмних продуктів.

13) Опрацювавши джерела 6-7, 9, 11, 18, 22, дати відповідь на питання: що таке валідація веб-сторінки? Яким чином перевірити сторінку на валідність?

14) Описати призначення та основні атрибути тега <meta>. Як вплине присутність такого тега на індексування сторінки інформаційно-пошуковою системою? Для підготовки відповіді використати джерела 5, 6, 18, 21 із рекомендованого списку.

15) Для чого призначений елемент <!doctype>? Чи впливає відсутність такого тега на коректність відображення сторінки браузером? Для підготовки відповіді використати джерела 6, 14, 18, 22 із рекомендованого списку.

16) Використавши джерела 8, 11, 15, 19, з'ясувати, якими були передумови переходу на протокол IP версії 6 з IPv4?

17) Що таке Веб 1.0, Веб 2.0, Веб 3.0? Навести їх основні характеристики та відмінності. Для підготовки відповіді використати джерела 8, 9, 11, 16, 17, 20 із рекомендованого списку.

## Завдання для індивідуальної роботи

**Завдання 1.** Підготувати виступ-повідомлення про один із програмних продуктів для роботи в Інтернет, в якому висвітлити питання:

- призначення програмного продукту;
- історія розвитку, версії, виробник;
- функціональні можливості;
- основні режими функціонування;
- особливості, переваги, недоліки;
- платний/безкоштовний, де можна придбати.

Доповідь обов'язково має супроводитися презентацією.

Перелік програмних продуктів, які виносяться на розгляд:

1. Mozilla Firefox +
2. Opera +
3. Google Chrome +
4. Internet Explorer +
5. Safari +
6. OUTLOOK EXPRESS
7. THE BAT!
8. ICQ (QIP)
9. Skype
10. Download Master
11. FlashGet
12. BitTorrent
13. Kerio Winrout
14. KIS

**Завдання 2.** Створити засобами HTML та CSS тематичний веб-сайт, який складається мінімум з 5 веб-сторінок, на яких відобразити форматований текст, списки, таблиці, вставити малюнки, гіперпосилання, закладки.

### Список рекомендованих джерел

1. Быстро и легко осваиваем работу в сети Интернет / [Под ред. Резникова Ф. А.] — М. : Лучш. кн., 2004. — 384 с.
2. Грязнова О. В. Основы Web-проектирования [Электронный ресурс] — Режим доступа : <http://matphys.rpd.univ.kiev.ua/ukr/courses/webdes.html>.
3. Лабораторный практикум по Web-технологиям. Часть 1. Основы HTML технологий. Лабораторный практикум / [Тубольцев М. Ф., Путивцева Н. П., Гурьянова И.В., Немикина О. В.] — Белгород : Белгородский государственный университет, 2003. — 49 с.
4. Гусев В. С. Аналитика веб-сайтов: использование аналитических инструментов для продвижения в Интернет / Гусев В. С. — М. : Диалектика, 2008. — 176 с.
5. Гусев В. С. Поиск в Internet: Самоучитель / Гусев В. С. — М. : Вильямс, 2004. — 324 с.
6. Для тех, кто делая сайты [Электронный ресурс] — Режим доступа : <http://htmlbook.ru/>.
7. Кемпбел М. Строим Web-сайты. Дизайн-HTML-CSS / Кемпбел М.; пер. с англ. А. Н. Горлач, А. В. Климович. — М. : Изд-во ТРИУМФ, 2006. — 480 с.
8. Кроудер Д. Создание веб-сайта / Кроудер Д.; пер. с англ. Е. Д. Михновой, В. С. Иващенко.— М. : Диалектика, 2010. — 360 с.
9. Левин Д. Интернет для "чайников"/Левин Д., Левин-Янг М., Бароди К.; пер. с англ. В. С. Гусева. — М. : Диалектика, 2010. — 352 с.
10. Левин М. Д. Методы поиска в Интернет / Левин М. Д. — М. : Солон-Пресс, 2009. — 224 с.
11. Матросов А. В. HTML 4.0 / Матросов А. В., Сергеев А. О., Чаунин М. П. — СПб. : БХВ-Петербург, 2007. — 672 с.
12. Остропицкий В. М., Мозгова И. В. Internet – технологии: Учебное пособие / В. М. Остропицкий, И. В. Мозгова. — Д. : Инновация, 2010. — 192 с.
13. Попов В. О. Практикум по Интернет-технологиям / Попов В. О. — С.Пб. : Питер, 2002. — 480 с.

14. Соломенчук В. Г. Интернет: Пособие для ускоренного обучения / Соломенчук В. Г. — С.Пб. : Питер, 2000. — 288 с.
15. Финков М. В. Интернет. Шаг второй: от пользователя к профессионалу / Финков М. В. — Спб. : Наука и Техника, 2002. — 768 с.
16. Харли Х. Эффективный самоучитель работы в Internet / Харли Х.; пер. с англ. А. Н. Горлач. — К. : Диасофт, 2001. — 448 с.
17. Шаров Е. Н. МЕТА-теги: описание и правила создания [Электронный ресурс] — Режим доступа : <http://www.gcmsite.ru/?pg=art&id=meta-keywords-create>.

## Додаток А

Таблиця спеціальних символів HTML

Ім'я	Код	Вид	Опис
&nbspsp	&#160		нерозривний пробіл
&pound	&#163	£	фунт стерлінгів
&euro	&#8364	€	знак євро
&para	&#182	¶	символ параграфа
&sect	&#167	§	параграф
&copy	&#169	©	знак copyright
&reg	&#174	®	знак зареєстрованої торгової марки
&trade	&#8482	™	знак торгової марки
&deg	&#176	°	градус
&plusmn	&#177	±	плюс-мінус
&frac14	&#188	¼	дріб – одна четверта
&frac12	&#189	½	дріб – одна друга
&frac34	&#190	¾	дріб – три четвертих
&times	&#215	×	знак множення
&divide	&#247	÷	знак ділення
&fnof	&#402	<i>f</i>	знак функції
<b>Грецькі букви</b>			
&Alpha	&#913	Α	грецька велика буква альфа
&Beta	&#914	Β	грецька велика буква бета
&Gamma	&#915	Γ	грецька велика буква гамма
&Delta	&#916	Δ	грецька велика буква дельта
&Epsilon	&#917	Ε	грецька велика буква епсилон
&Zeta	&#918	Ζ	грецька велика буква дзета
&Eta	&#919	Η	грецька велика буква ета
&Theta	&#920	Θ	грецька велика буква тета
&Iota	&#921	Ι	грецька велика буква йота
&Kappa	&#922	Κ	грецька велика буква каппа
&Lambda	&#923	Λ	грецька велика буква лямбда
&Mu	&#924	Μ	грецька велика буква мю
&Nu	&#925	Ν	грецька велика буква ню
&Xi	&#926	Ξ	грецька велика буква ксі
&Omicron	&#927	Ο	грецька велика буква омікрон
&Pi	&#928	Π	грецька велика буква пі
&Rho	&#929	Ρ	грецька велика буква ро
&Sigma	&#931	Σ	грецька велика буква сигма
&Tau	&#932	Τ	грецька велика буква тау
&Upsilon	&#933	Υ	грецька велика буква іпсилон
&Phi	&#934	Φ	грецька велика буква фі



&Chi	&#935	Χ	грецька велика буква χі
&Psi	&#936	Ψ	грецька велика буква псі
&Omega	&#937	Ω	грецька велика буква омега
&alpha	&#945	α	грецька мала буква альфа
&beta	&#946	β	грецька мала буква бета
&gamma	&#947	γ	грецька мала буква гамма
&delta	&#948	δ	грецька мала буква дельта
&epsilon	&#949	ε	грецька мала буква епсилон
&zeta	&#950	ζ	грецька мала буква дзета
&eta	&#951	η	грецька мала буква ета
&theta	&#952	θ	грецька мала буква тета
&iota	&#953	ι	грецька мала буква йота
&kappa	&#954	κ	грецька мала буква каппа
&lambda	&#955	λ	грецька мала буква лямбда
&mu	&#956	μ	грецька мала буква мю
&nu	&#957	ν	грецька мала буква ню
&xi	&#958	ξ	грецька мала буква ксі
&omicron	&#959	ο	грецька мала буква омікрон
&pi	&#960	π	грецька мала буква пі
&rho	&#961	ρ	грецька мала буква ро
&sigmaf	&#962	ς	грецька мала буква сигма
&sigma	&#963	σ	грецька мала буква сигма
&tau	&#964	τ	грецька мала буква тау
&upsilon	&#965	υ	грецька мала буква іпсилон
&phi	&#966	φ	грецька мала буква фі
&chi	&#967	χ	грецька мала буква χі
&psi	&#968	ψ	грецька мала буква псі
&omega	&#969	ω	грецька мала буква омега
Стрілки			
&larr	&#8592	←	стрілка вліво
&uarr	&#8593	↑	стрілка вверх
&rarr	&#8594	→	стрілка вправо
&darr	&#8595	↓	стрілка вниз
&harr	&#8596	↔	стрілка вліво-вправо
Інші символи			
&spades	&#9824	♠	
&clubs	&#9827	♣	
&hearts	&#9829	♥	
&diams	&#9830	♦	
&quot	&#34	"	подвійна лапка
&amp	&#38	&	амперсанд
&lt	&#60	<	знак "менше"
&gt	&#62	>	знак "більше"

Знаки пунктуації			
&hellip	&#8230	...	...
&prime	&#8242	'	мінUTI і фути
&Prime	&#8243	"	секунди і дюйми
Загальна пунктуація			
&ndash	&#8211	–	тире
&mdash	&#8212	—	довге тире
&lsquo	&#8216	‘	
&rsquo	&#8217	’	
&sbquo	&#8218	‚	
&ldquo	&#8220	“	
&rdquo	&#8221	”	
&bdquo	&#8222	„	
&laquo	&#171	«	
&raquo	&#187	»	