**ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Практикум містить систематизований курс лабораторних робіт з розділу дисципліни “WEB-технології та WEB-дизайн”, що вивчає основи верстання web-документів засобами мови HTML і CSS для студентів спеціальності 122 “Комп’ютерні науки”.

Базовим програмним забезпеченням даного розділу курсу є Denwer і HTMLPad. Denwer - це набір дистрибутивів та програмна оболонка, що використовуються Web-розробниками для відладки сайтів без необхідності виходу до Інтернету, а HTMLPad – багатофункціональний редактор HTML з підсвічуванням синтаксису і попереднім переглядом документа. Обидва вибрані програмні продукти містять довідкову систему з основних тем курсу, що дозволяє удосконалити навички з пошуку інформації.

В лабораторному практикумі розглядаються такі питання, як: встановлення програмних продуктів, структура web-сервера, структура html-документа, основні теги форматування тексту, абзаців, зображень, гіперпосилань, списків, таблиць, форм, фреймів, карт зображень,підключення таблиць стилів до документа, можливості CSS по роботі з блоковими об'єктами, структура таблиці для прискорення створення стилів CSS, можливості роботи зі списками за допомогою CSS, приклади верстання веб-сторінок та ін.

До кожної лабораторної роботи додається архів з необхідними для виконання робіт даними.

Після вивчення курсу лабораторних робіт студентам надається завдання для самостійної роботи: створення статичного сайту із застосуванням отриманих знань. У ДОДАТКУ Г наведено частину варіантів завдань (повний перелік надає викладач).

Вимоги до апаратного і програмного забезпечення:

Для виконання завдань лабораторних робіт необхідно мати комп'ютер, який має оперативну пам'ять не менше 256 Кб і жорсткий диск, з вільним об'ємом не менше 1 Гб. Операційна система – Windows XP і більш пізні версії, включаючи повну сумісність програмного забезпечення з Windows Vista.

**1 ЛАБОРАТОРНА РОБОТА №1**

| **Тема:** | **Web-сервер Denwer. Введення в HTML** |
| --- | --- |
| **Мета:** | 1. Навчитися встановлювати Denwer, створювати власні домени і піддомени в програмі Denwer. 2. Освоїти основні принципи роботи з програмою HTMLPad; 3. Навчитися застосовувати теги HTML для структурування сторінки і форматування тексту. |
| **Час:** | 2 год. |

**1.1 Теоретичні відомості**

1.1.1 Denwer. Встановлення програмного продукту.

Програма Denwer призначена для створення власних доменів на локальному комп'ютері, що дозволяє створювати сайти і тестувати їх працездатність, не маючи доступу до Інтернету.

Для встановлення програмного продукту Denwer необхідно викачати дистрибутив з сайту http://denwer.ru і запустити установний файл. Під час встановлення необхідно вибрати наступні опції:

1. встановлення здійснити в папку C:\WebServer, а не в звичну папку C:\Program Files;
2. вибрати варіант запуску вручну, а не як службу.

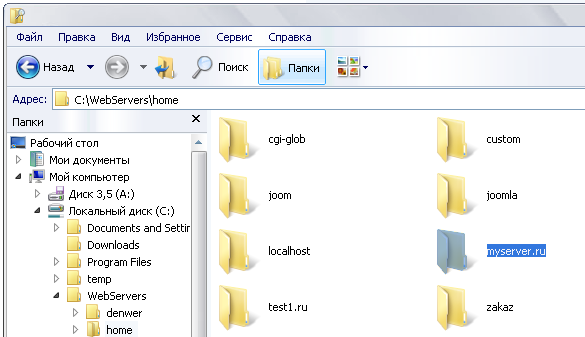
Після встановлення Денвера його слід запустити вручну і перевірити працездатність. Для цього в браузері необхідно набрати адресу тестової сторінки http://localhost/denwer/. В разі успішного встановлення на екрані з'явиться сторінка «Ура заработало!».

1.1.2 Створення власного домену на локальному комп'ютері

Після встановлення програми всі необхідні для її працездатності файли розташовуються в папці C:\WebServers. Завдяки цьому папку можна переносити з комп'ютера на комп'ютер, не порушуючи при цьому працездатність сервера.

Для створення власного домену в межах локального комп'ютера досить додати нову папку з ім'ям домену в каталог C:\WebServers\home. Наприклад myserver.ru (див. рис.1.1).

Файли, які повинні знаходитися на цьому сервері необхідно помістити в папку C:\WebServers\home\myserver.ru\www. При створенні власного домену слід враховувати неприпустимість використання в імені домену пробілів, букв кирилиці і спеціальних символів.



**Рис.1.1. Розміщення доменів на Denwer**

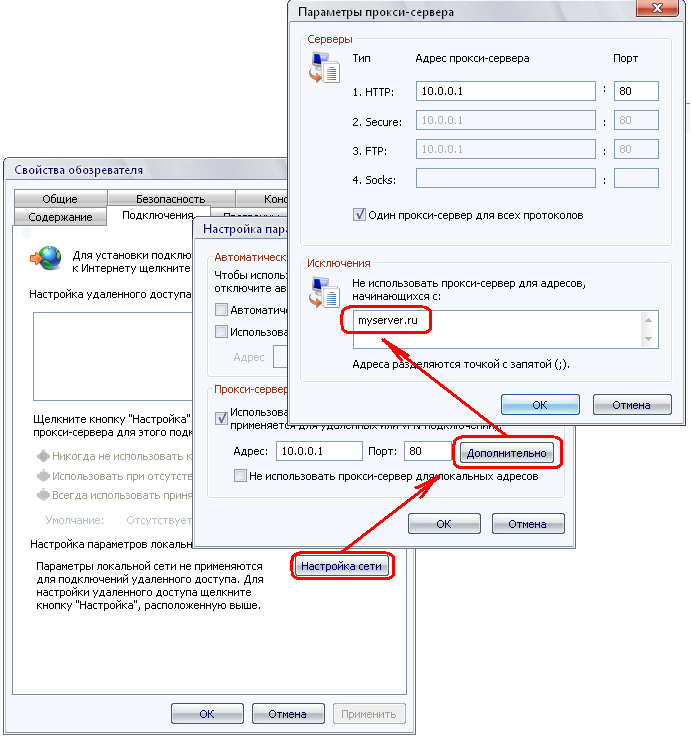
Для перевірки працездатності нового домену слід:

а) перевантажити Denwer (зробити це можна за допомогою ярлика  Restart servers в меню Пуск або за допомогою файлу запуску C:\WebServers\Denwer\Restart.exe);

б) відкрити в браузері сторінку з адресою нового домена (наприклад myserver.ru). При цьому в разі успішного запуску сервера на екрані відобразиться список файлів або зміст файлу з ім'ям index. На більшості сайтів при зазначені шляху до домену або до папки відображується файл index.html або index.php або вміст каталога (у випадку якщо індексний файл відсутній).

Варіант коли новий піддомен не відображується в браузері:

1. виконати команду ping на новий домен (ping myserver.ru). Якщо відповідь отримана, це означає що Denwer з новими налаштуваннями успішно запущений. Якщо відповіді немає, то можливо а) Денвер не був перезапущений з моменту додавання нового домену; б) ім'я теки відрізняється від імені пінгуємого домену; в) всередині домену не створена папка www.
2. якщо команда ping проходить, то можливо в браузері використовується проксі-сервер для доступу до Інтернету (див. рис.1.2). Наприклад в браузереі Internet Explorer необхідно додати ваш домен в список винятків. Для цього необхідно зайти в Сервис ⇒ Свойства обозревателя ⇒ Подключение ⇒ Настройка сети ⇒ Дополнительно... у полі Исключения вказати адресу вашого нового домену.
3. ще раз перевірте що ім'я домену містить лише латинські літери, цифри та не містить пробілів.



**Рис.1.2. Додавання винятків для проксі-сервера**

1.1.3 Інтерфейс програми HTML Pad

Після встановлення сервера і створення власного домену можна приступати до створення сайту. Сайти можна умовно розділити на статичні (будуть доступні користувачеві в тому вигляді і з тією інформацією якою вони були створені) і динамічні (наповнення і зовнішній вигляд сторінок може залежати від ряду умов: облікового запису користувача, часу та інше).

Статичні сайти є набором html-сторінок, а динамічні - набір php-сторінок. Статичний сайт повинен містити html-сторінки і картинки розташовані в одній теці. Створювати такі сторінки можна або в блокноті, або в спеціалізованих програмах. Однією з таких програм є програма HTMLPad. В порівнянні з блокнотом в програмі є підсвічування синтаксису, автоматичне доповнення, панель властивостей, попередній перегляд та інше.

На рис.1.3 відображують основні елементи інтерфейсу вікна програми “HTMLPad 2008”. Робота з файлами (їх створення, збереження, відкриття) властива всім додаткам Windows, з тією лише різницею, що при створенні нового документа, необхідно вибирати із списку саме HTML Page.

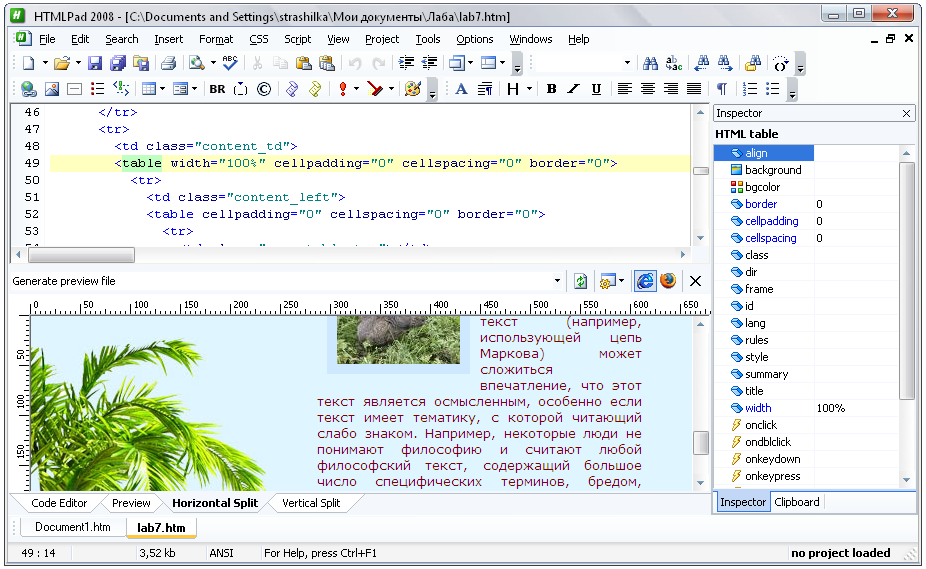
******

Рис.1.3. Інтерфейс програми “ HTML Pad 2008”

Після створення нової сторінки основні теги можна додавати вручну, або через пункти меню Insert (Вставити) і Format (Форматування), або за допомогою кнопок на панелях інструментів HTML, HTML Formatting та HTML Forms.

Перемикатися між режимом редагування і режимом попереднього перегляду можна також за допомогою використання вкладок перемикання режимів перегляду (див. рис.1.3).

1.1.4 Структура HTML-документа

Складання сторінок для сайту базується на мові HTML (від англ. Hypertext Markup Language – мова розмітки гіпертексту). Всі сторінки сайту мають розширення **.htm** або**.html.**

Основною структурною одиницею мови є тег з його атрибутами. Тег являє собою розміщені в трикутних дужках інструкції мови, які повідомляють, якими властивостями повинен володіти той або інший фрагмент тексту на сторінці. Більшість тегів є парними: містять як відкриваючий, так і закриваючий тег (див. рис.1.4):

**Рис.1.4. Структура тега**

Слід зазначити, що написання значень атрибутів потрібно писати в подвійних лапках.

Опис всього документа будується з тегів. Весь документ має бути розміщений в головному тегу **<html>** (див. рис.1.5).



<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=windows-1251"/>

<title> Заголовок документа </title>

<link href="style.css" rel="stylesheet" type="text/css"/>

</head>

<body>

<p><b>Важлива</b> частина документу...</p>

</body>

</html>

**Рис.1.5. Структура html-документа**

Усі теги можна умовно розділити на групи по їх функціональному призначенню: теги рівня документа, теги форматування, теги структурних елементів та ін.

1.1.5 Теги рівня документа

**<html>...</html>** парний тег-контейнер, який містить в собі всю веб-сторінку. Відкриваючий та закриваючий теги **<html>** в документі обов'язкові, для того щоб вказати межі документа.

**<head>...</head>** тег призначений для зберігання елементів, які використовуються для зберігання службової інформації, призначеною для браузерів і пошукових систем. Вміст тега не відображується на сторінці, за винятком тега **<title>.**

**<title>...</title>** містить текст заголовка, який відображується в рядку заголовка вікна браузера.

**<body>...</body>** призначений для зберігання вмісту веб-сторінки (контента), браузера, що відображується у вікні (форматований текст,зображення, таблиці і так далі). Тег **<body>** може містити безліч атрибутів, які впливають на відображення всього документа вцілому.

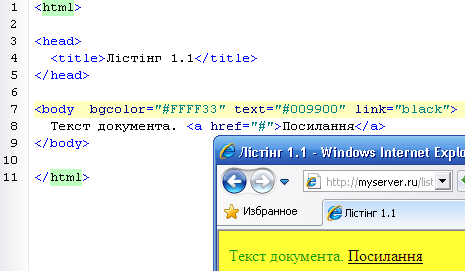
Таблиця 1.1

**Основні атрибути тега <body>**

| **Назва атрибуту** | **Призначення атрибуту** |
| --- | --- |
| alink | колір активного посилання (колір задається шістнадцятиричним числом або константою, наприклад червоний: red або #ff0000) |
| background | фоновий рисунок на веб-сторінці |
| bgcolor | колір фону веб-сторінки |
| topmargin(leftmargin, bottommargin) | відступ від верхнього (лівого, нижнього) краю вікна браузера до контента |
| link | колір посилань на веб-сторінці |
| text | колір тексту в документі |
| vlink | колір відвіданих посилань |

Наприклад, наступний фрагмент html-кода задає для всього документа колір тексту зелений, колір посилань чорний, колір фону жовтий. Лістинг 1.1.

**Лістинг 1.1 – Приклад використання атрибутів тега <body>**



1.1.6 Основні теги форматування тексту

Таблиця 1.2

**Теги форматування тексту**

| **Назва тега** | **Призначення тега, атрибути** |
| --- | --- |
| 1 | 2 |
| <b>...</b>  або  <strong>...</strong> | встановлює жирне зображення шрифту |
| <i>...</i> | встановлює курсивне зображення шрифту |
| <u>...</u> | встановлює підкреслене зображення шрифту |
| <strike>...</strike> | встановлює закреслене зображення шрифту |
| < cite>...</cite> | позначає текст як цитату, зазвичай відображується курсивом |

Продовження таблиці 1.2

| 1 | 2 |
| --- | --- |
| <code>...</code> | призначений для відображення тексту, який є програмним кодом, зазвичай відображується моноширинним шрифтом |
| <del>...</del> | використовується для виділення тексту, який був видалений в новій версії документа |
| < ins>...</ins> | використовується для виділення тексту, який був доданий в нову версію документа |
| <dfn>...</dfn> | застосовується для виділення термінів при їх першій появі в тексті |
| <sup>...</sup> | відображує шрифт у вигляді верхнього індексу |
| <sub>...</sub> | відображує шрифт у вигляді нижнього індексу |
| <pre>...</pre> | визначає блок тексту, в якому зберігається кількість пропусків між словами, задана на етапі створення документа |
| < nobr>...</nobr> | повідомляє браузер відображувати текст в один рядок без перенесення |
| <h1>...</h1>  ...  <h6>...</h6> | встановлює заголовки різного рівня значущості від найкрупнішого h1 до найдрібнішого h6 |
| <font>...</font> | призначений для встановлення характеристик шрифту, які задаються в атрибуті тега.  Тег має наступні атрибути:  **color=…** задає колір шрифту (у шістнадцятиричній формі, наприклад, #ff0000)  **face=…**задає гарнітуру шрифту(Arial,Tahoma)  **size=…** задає розмір шрифту в умовних одиницях (ціле число від 1 до 7 або зміни числа +1 або -2) найкрупніший шрифт 7 |
| <p>...</p> | визначає межі абзацу. Тег має атрибут **align,** який відповідає за вирівнювання тексту і може набувати значень: **left, right, center**, **justify** |
| <br> | встановлює примусове перенесення рядків |
| < hr> | малює горизонтальну лінію. Параметри горизонтальної лінії можна задати з використанням атрибутів тега:  **align=…** визначає вирівнювання лінії  **color=…** колір лінії  **noshade=…** малює не тривимірну лінию  **size=…** товщина лінії в пікселях  **width=…** ширина лінії в пікселях |

При встановленні розмітки тексту теги можна писати вручну, або користуватися панелями інструментів HTML, HTML Formatting, HTML Tables і HTML Forms. Наприклад, для встановки жирного зображення тексту слід:

1. виділити фрагмент тексту, який слід зробити жирним;
2. натиснувати на панелі інструментів HTML Formatting кнопку з вказівкою жирного зображення;
3. в результаті необхідні теги будуть додані в документ;
4. якщо є необхідність задати атрибути тега, то зробити це можна за допомогою палітри Inspector. Виділити тег, якому слід змінити атрибути і встановити їх значення.

1.1.7 Спеціальні символи і коментарі

Ряд символів в мові використовується в службових цілях (знаки більше і менше), а ряд символів неможливо набрати на клавіатурі (нерозривний пробіл, копірайт). Для відображення таких символів використовують спеціальні коди, які пишуть під час верстки html-сторінок (див. Таблиця 1.3):

Таблиця 1.3

**Спецсимволи мови HTML**

| **Ім'я** | **Код** | **Вигляд** | **Опис** |
| --- | --- | --- | --- |
| &quot; | &#34; | " | подвійна лапка |
| &amp; | &#38; | & | амперсанд |
| &lt; | &#60; | < | знак 'менший' |
| &gt; | &#62; | > | знак 'більший' |
| &nbsp; | &#160; |  | нерозривний пропуск |
| &sect; | &#167; | § | параграф |
| &сміттю; | &#169; | © | знак copyright |
| &reg; | &#174; | ® | знак зареєстрованої торгівельної марки |
| &deg; | &#176; | ° | градус |
| &plusmn; | &#177; | ± | плюс-мінус |
| &trade; | &#8482; | ™ | торгівельна марка |
| &deg; | &#176; | ° | градус |

Такі символи можна додавати як у вигляді імені, так і у вигляді коду. У програмі HTMLPad для цього можна використовувати команду Insert ⇒ Character ⇒ Special Character.

Для коментарів або тимчасового виключення тексту з html-сторінки використовують пари символів: **<!--Текст коментаря-->.**

**1.2 Практична частина**

1. Запустіть веб-сервер: Пуск ⇒ Денвер ⇒ Start (або C:\WebServers\denwer\run.exe). Коли він запуститься в системному треї, у вас з'явиться значок у вигляді малинового пера.
2. Перевірте роботу Денвера виконавши в командному рядку команду **ping test1.ru**. Якщо відповідь походить від адреси 127.0.0.1, то це означає, що Денвер запущений і успішно працює на локальному комп'ютері.
3. Відкрийте в браузері адресу **test1.ru**. Якщо відбувається запит пароля, то відключіть в налаштуваннях проксі сервер для адреси **test1.ru**. Для Internet Explorer необхідно виконати наступні дії: Сервіс ⇒ Властивості оглядача ⇒ Підключення ⇒ Налаштування мережі ⇒ Додатково. У полі Не використовувати проксі сервер для адрес тих, що починаються з: додати до наявних записів (через крапку з комою) адресу **test1.ru**.
4. Оновіть поточну сторінку з адресою **test1.ru** (натисніть F5).
5. Для створення власних доменів, на локальному комп'ютері створіть в папці c:\WebServers\home папку з ім'ям домену **help.ua** або **help.ru**, на якому ви будете проводити тестування сайту. Сам сайт знаходиться в архіві, що додається до лабораторної роботи.
6. У папці **help.ua** (або **help.ru**) створіть папку **www** , в яку скопіюйте вміст папки **htmlbook** (узятою з архіву до лабораторної роботи).
7. Перевірте щоб файл **index.html** містився в папці **www**, а не у вкладених папках.
8. Використовуючи меню Пуск перезапустіть Денвер (для того щоб знов створений домен “піднявся”). Перевірте працездатність сайту через команду **ping**, а потім подивіться в браузері. Зверніть   
   увагу, що на екрані відображуються файли з папки c:\WebServers\home\help.ua\www, але шлях до файлів, вказанний як для сервера, з вказівкою доменного імені, а не як для файлів з диска С.
9. Створіть свій домен, в імені якого буде присутнє ваше прізвище і група. У папці **www** вашого каталога створіть файл **index.html.** Саме з цього файлу завжди починається завантаження вмісту домену, якщо не вказанний інший файл.
10. Відкрийте файл **index.html** за допомогою програми HTML Pad 2008.
11. Запишіть в ньому наступні рядки:

<html>

<head>

<title> Заголовок вікна </title>

</head>

<body>

Освоюємо web-технології :)

</body>

</html>

1. Збережіть файл. Перезапустіть Денвер (веб-сервер). Перевірте в браузері результат, набравши в адресному рядку ім'я вашого сервера із завдання 9. У разі виникнення проблем із завантаженням сервера зверніться до теоретичних відомостей.
2. Зверніть увагу на структуру вікна в програмі HTML Pad 2008. У правій частині вікна знаходиться Інспектор атрибутів тегів (якщо вікна Inspector немає на екрані, то відобразити його можна командою View ⇒ Inspector). Це вікно містить додаткові атрибути тегів (ім'я тега завжди розташовується між трикутною парою дужок).
3. Встановіть курсор усередині тега **<body>** та прогляньте які атрибути є в цього тега. Змініть атрибути **text** (білий) і **bgcolor** (синій). Збережіть зміни. Перегляньте результат на вкладці Preview і в браузері. Таким чином, додаючи теги і змінюючи їх атрибути можна створити документ будь-якої складності.
4. Всі файли і зображення, які з'являтимуться на вашій веб-сервері-сторінці, повинні розташовуватися всередині папки **<ім'я домена>/www** і мати адреси починаючи з кореня сайту, а не з кореня диска С: або D: Виберіть фонове зображення, скопіюйте його в папку **www** і змініть фон сторінки (атрибут **background** в тезі **body**). Збережіть і прогляньте результати роботи в браузері.
5. Використовуючи інформацію текстового файлу з архіву до лабораторної роботи відформатуйте вміст сторінки таким чином, щоб вона була схожа на рис.1.6 (детальний опис починається з пункту 17). При форматуванні використовуйте теги з теоретичних відомостей, а також детальний опис тегів з домена **html.ua(.ru);**
6. Створіть новий документ з ім'ям **lab1.html** і збережіть його в корені свого домену.
7. Між відкриваючим та закриваючим тегом **<body>** додайте текст з файлу. Збережіть його і прогляньте в браузері що вийшло.
8. Змініть фон сторінки і колір тексту згідно рисунку.



**Рис.1.6. Зразок для виконання практичного завдання**

1. Розбийте текст на абзаци використовуючи тег **<p>**.
2. Зверніть увагу на те, щоб у вас не було червоних (не закритих) тегів.
3. Встановіть для абзаців вирівнювання. Для цього встановіть курсор всередину тега абзацу, що відкривається (у тег **<p>**). Перейдіть на панель Inspector і в атрибуті **aling** виберіть потрібне значення. Якщо значення вирівнювання по ширині не вибирається із списку, то встановіть його вручну: **align=“justify**”.
4. Встановіть словам жирне, підкреслене і курсивне накреслення.
5. У другому абзаці використайте верхній індекс.
6. Використовуючи спеціалізовані теги виділіть цитату і програмний код. Використовуйте додатковий тег **<pre>** для збереження числа пропусків між словами в програмному коді.
7. Використовуючи тег **<font>** виділіть назви пошукових систем.
8. Змініть колір яндекса на жовтий.
9. Додайте в документ дві лінії, першу завдовжки в 400 пікселів, а заввишки в 5 пікселів. А другу і третю шириною 50%.
10. Використовуючи теги **<h1>...<h6>** виділіть заголовки.
11. Використовуючи коди спеціальних символів відобразіть на сторінці знак менше. І підпишіть виконане завдання як вказано на рисунку 1.6.
12. Створіть новий файл **lab1.1.html** і відформатуйте текст таким чином, щоб він відображався в браузері як на рисунку **code.png** з архіву до лабораторної роботи.
13. Перевірте, що всі файли у вас доступні через сервер з використання доменного імені.
14. Покажіть виконану роботу викладачу.

**1.3 Контрольні питання**

1. Як додати до Denwer свій домен?
2. Як перевірити працездатність нового домену?
3. Яку програму можна використовувати для створення html-сторінок?
4. Для чого призначена палітра Inspector в програмі HTMLPad 2008?
5. Що таке тег?
6. Які існують теги рівня документа?
7. Якими тегами можна виділити текст жирним, курсивом, підкресленням?
8. Які атрибути використовуються тегом **<body>**?
9. Якими тегами можна відзначити верхній і нижній індекси?
10. Який тег дозволяє змінити одночасно шрифт, колір тексту і розмір тексту?
11. У чому відзнака тегів **<br>** і **<p>**?
12. За допомогою чого можна управляти вирівнюванням абзацу?
13. Як додати в документ горизонтальну лінію?
14. Що таке спеціальні символи і як їх можна додати в документ?
15. Між якими символами розташовується текст коментарів?

**2 ЛАБОРАТОРНА РОБОТА №2**

| **Тема:** | **Зображення. Списки. Посилання** |
| --- | --- |
| **Мета:** | 1. Навчитися додавати зображення в документ і управляти атрибутами тега <img>; 2. Навчитися створювати нумеровані, маркеровані і багаторівневі списки; 3. Навчитися створювати внутрішні і зовнішні посилання. |
| **Час:** | 2 год. |

**2.1 Теоретичні відомості**

2.1.1 Зображення

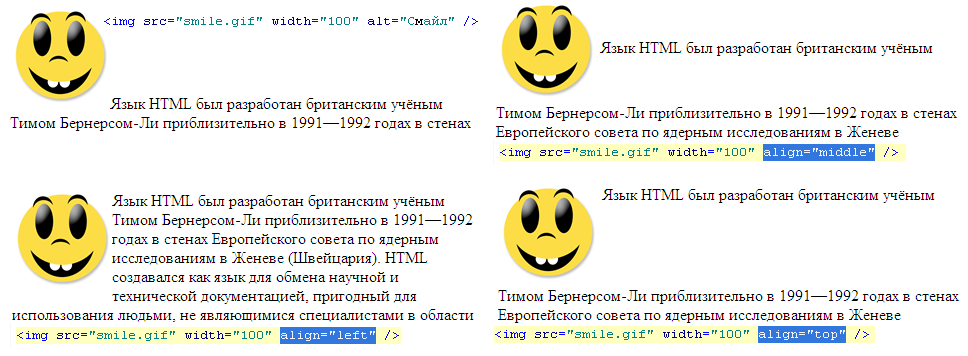
Вставка картинок в html-документ здійснюється з використанням тега **<img>.** Тег можна записати вручну і потім за допомогою автодоповнення коду або за допомогою панелі Inspecrot вказати його атрибути. Можна також скористатися кнопкою Image на панелі інструментів HTML програми HTMLPad 2008.

Тег **<img>** має наступні атрибути (див. таблицю 2.1):

Таблиця 2.1

**Атрибути тега <img>**

| **Атрибут** | **Призначення** |
| --- | --- |
| src | вказує на напрямок до графічного файлу. Напрямки можуть бути відносними: photo.jpg, ../img/photo2.gif або абсолютними: http://mysite.com/pic/photo3.jpg |
| lowsrc | напрямок до графічного файлу гіршої якості (і меншого розміру), який вантажиться перед повнорозмірною картинкою |
| alt і title | альтернативний текст для зображення, використовується як спливаюча підказка або пошуковими системами |
| border | товщина рамки довкола зображення. Колір рамки визначається поточним кольором тексту |
| hspace | горизонтальний відступ від зображення до тексту |
| vspace | вертикальний відступ від зображення до тексту |
| height | висота зображення, якщо не задана, то відображується оригінальний розмір зображення, якщо вказана до зображення масштабується |
| width | ширина зображення, якщо при вказівці ширини одночасно з висотою не зберігаються пропорції, то зображення спотворюється |
| align | визначає як малюнок буде вирівнюватися і обтікатися текстом (див. рис.2.1) |

****

**Рис.2.1. Приклади вирівнювання зображення відносно тексту**

2.1.2 Списки

У мові HTML розрізняють наступні види списків:

а) маркіровані списки;

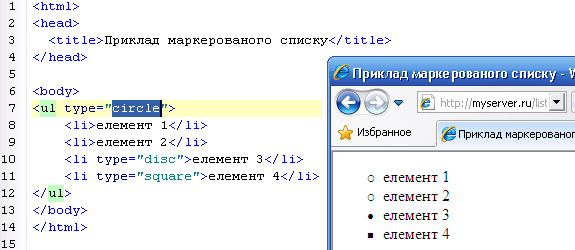
б) нумеровані списки;

в) списки визначень.

Для створення маркірованих списків застосовують теги **<ul>** і **<li>.** Тегом **<ul>** наголошується початок і закінчення всього списку. Тегом **<li>** позначають початок і кінець окремого елементу списку.

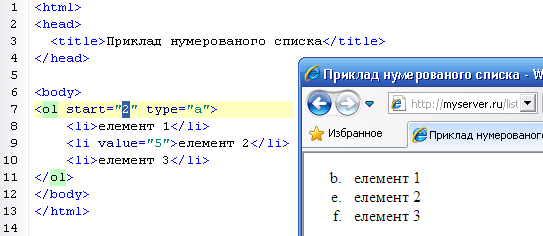
За замовчуванням елементи списку маркіруються чорним кружечком. За допомогою атрибуту **type** можна змінити стиль маркірування. Приклад використання списку наведений в Лістингу 2.1:

**Лістинг 2.1 – Приклад маркірованого списку**



Аналогічно маркірованих списків, використовуючи теги **<ol>** і **<li>** можна створювати нумеровані списки. Для створення варіативної нумерації використовують атрибут **type** для вибору стилю нумерації і **start** для вказівки того, з якого символу слід почати нумерацію списку. Для тега **<li>** атрибут **value** дозволяє задати номер поточному елементу списку.

**Лістинг 2.2 – Приклад використання нумерованого**



Списком визначень є текст, що складається з двох взаємопов'язаних наборів – списку з термінами і списку визначень термінів. Спочатку вказується перший термін, нижче за нього йде його визначення, потім наступний термін з визначенням і так далі

Структура списку визначень наступна:

***Термін 1***

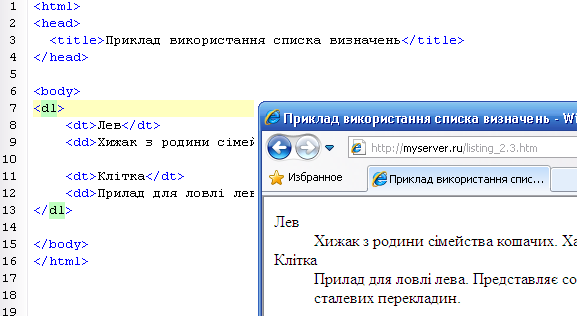
Визначення терміну 1

***Термін 2***

Визначення терміну 2

Сам список задається за допомогою тега **<dl>,** термін – тегом **<dt>,** а його визначення – за допомогою тега **<dd>**. Вкладення тегів для створення списку визначень продемонстроване в лістингу 2.3:

**Лістинг 2.3 – Приклад використання списку визначень**



2.1.3 Посилання

***Посилання або Гіперзв'язок*** *(Link, Hyperlink)* – фрагмент тексту або графіки на HTML-сторінці що посилається на іншу позицію в тому ж документі або на об'єкт в іншому документі (можливо навіть розташованому на іншому сервері).

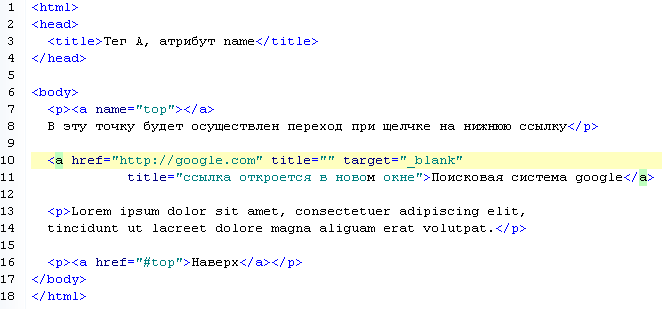
Для створення посилань в мові HTML використовують тег **<a>** (anchor – якір). Атрибути, які використовуються в посиланні приведені в таблиці 2.2.

Таблиця 2.2

**Атрибути тега <а>**

| **Атрибут** | **Призначення** |
| --- | --- |
| href | задає адресу документа, на який слід перейти. Є обов'язковим атрибутом для тега **<a>**. Може містити як відносну, так і абсолютну адресу сторінки. А також протоколи відмінні від http, наприклад ftp, mailto, ed2 та ін. |
| name | встановлює ім'я якоря, для визначення позиціювання посилання усередині документа. |
| target | встановлює ім'я вікна або фрейма, в якому буде завантажений документ. Можливі також варіанти використання: \_self для відкриття посилання в тому ж вікні і \_blank для відкриття посилання в новому вікні. |
| title | додає спливаючу підказку до тексту посилання |

**Лістинг 2.4 – Приклад використання посилань**



**2.2 Практична частина**

1. Запустіть веб-сервер Денвер.
2. Створіть в своєму домені файл lab2.htm.
3. Використовуючи дані з архіву до лабораторної роботи вставте в документ зображення з урахуванням: ширини, висоти, вирівнювання, спливаючих підказок та ін.
4. Створіть список визначень згідно малюнку. Використовуйте для цього теги **<dl>**, **<dd>** і **<dt>.**
5. Створіть маркірований список використовуйте тег **<ul>**. Зміните зовнішній вигляд маркера на квадратик.
6. Створіть нумерований список. Здійсніть нумерацію римськими цифрами і змініть початок нумерації з числа XV.
7. Створіть список з багаторівневою нумерацією.
8. Позначте початок кожного розділу сьогоднішньою лабораторною як іменований якір: **<а name=“image”>**, **<а name=“spisok”>**.
9. На початку сторінки додайте список посилань на всі розділи лабораторної роботи. Пам'ятаєте, що при посиланні на іменований якір слід використовувати знак # перед ім'ям якоря.
10. Додайте внизу сторінки посилання на завантаження інсталяційного файлу Денвера.
11. Додайте внизу документа посилання на свою сторінку в контакті, яка відкриватиметься в новому вікні.
12. Друге зображення в документі зробіть посиланням на пошукову систему Google
13. Створіть новий файл **index.html**, у якому розробіть список, що містить посилання на завдання двох лабораторних робіт.
14. Створіть новий файл **my.html** і запишіть в ньому: нумерований список з відомих вам мов програмування, маркірований список з відомих вам палітр кольорів.
15. Додайте посилання на файл **my.html** у файл **index.html**.
16. Покажіть виконану роботу викладачеві.

**2.3 Контрольні питання**

1. Який тег служить для вставки зображення в html-документ?
2. Де в тезі **<img>** вказується шлях до графічного файлу?
3. Як задати вирівнювання картинки відносно тексту?
4. Які списки існують в HTML?
5. Який тег бере участь в створенні як маркірованого так і нумерованого списків?
6. Як змінити порядок нумерації в нумерованому списку?
7. Як змінити вигляд маркера в маркірованому списку?
8. Що таке списки визначень?
9. Як задати якір для посилань в межах одного документа?
10. За що відповідає атрибут **target** в тезі **<a>**?
11. Як перевірити працездатність нового домену?

**3 ЛАБОРАТОРНА РОБОТА №3**

| **Тема:** | **Таблиці** |
| --- | --- |
| **Мета:** | 1. Навчитися створювати таблиці заданого розміру; 2. Навчитися об'єднувати клітинки по горизонталі і вертикалі; 3. Навчити створювати фіксовані і динамічні таблиці; 4. Навчитися управляти відстанню між клітинками і рамкою таблиці. |
| **Час:** | 2 год. |

**3.1 Теоретичні відомості**

3.1.1 Таблиці. Загальні відомості

Таблиці в HTML організовуються як набір стовпців і рядків. Елементи таблиці можуть містити будь-які HTML-елементи, такі, як заголовки, списки, абзаци, фігури, графіку, а також елементи форм. Таблиці з невидимою межою довгий час використовувалися для верстки веб-сторінок, дозволяючи розділяти документ на модульні блоки. Подібний спосіб вживання таблиць знайшов втілення на багатьох сайтах, але йому на зміну не прийшов сучасніший спосіб верстки за допомогою шарів.

Для створення таблиць користуються тегом **<table>,** який визначає початок і кінець таблиці. Основні атрибути тега **<table>** приведені в таблиці 3.1.

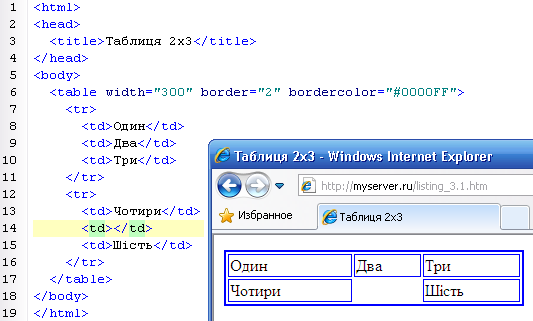
Таблиця 3.1

**Атрибути тега <table>**

| **Атрибут** | **Призначення** |
| --- | --- |
| аlign | визначає вирівнювання таблиці відносно всього документа |
| width | ширина таблиці у відсотках або в пікселях |
| height | висота таблиці. Часто не підтримується браузерами. У випадку якщо розмір тексту перевищує розмір таблиці – таблиця розтягується на висоту тексту |
| background | фоновий малюнок для всієї таблиці |
| bgcolor | задає колір фону таблиці |
| border | ширина рамки довкола таблиці, може бути значно товще ніж ширина рамки між клітинками |
| cellspacing | відстань між клітинками |
| cellpaddiing | відступ від рамки до вмісту клітинки |

При побудові таблиця розділяється на рядки (тег **<tr>**), які у свою чергу діляться на клітинки (тег **<td>**). Приклад таблиці наведений в лістингу 3.1.

**Лістинг 3.1 – Реалізації таблиці на два рядки і три стовпці**

****

У наданій на лістингу таблиці, за допомогою атрибутів встановлені фіксована ширина, рамка, шириною в два пікселі і колір рамки. При заповненні клітинок, якщо між парою тегів **<td>...</td>** немає графічних символів або спеціального символу нерозривний пробіл – межа клітинки на сторінці не відображатиметься.

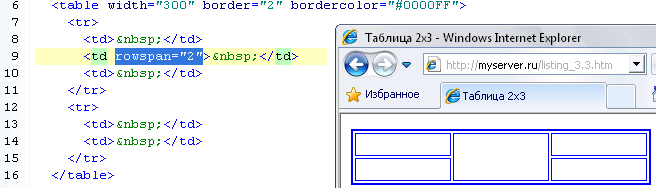
Замість тексту всередині клітинки може бути будь-який елемент: список, зображення, інша таблиця та ін.

Тег **<tr>** служить контейнером для створення рядка таблиці. Кожна клітинка в межах такого рядка може задаватися за допомогою тега **<th>** або **<td>.** Можливі атрибути тега **<tr>: align, valign**, **bgcolor, bordercolor**. Можливі атрибути для тегів **<td>** і **<th>:** **align, background, bgcolor, bordercolor, colspan, height, nowrap, rowspan, valign, width.**

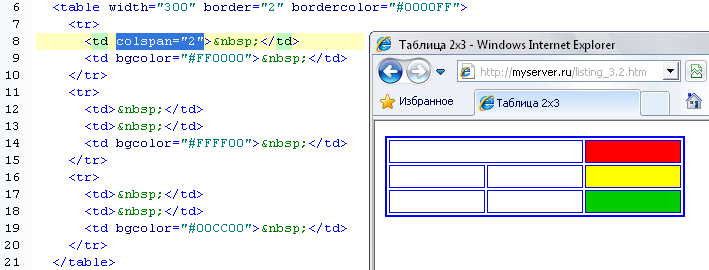
3.1.2 Об'єднання елементів таблиці

Для об'єднання елементів таблиці використовують атрибути colspan – для вказівки числа об'єднуваних клітинок по горизонталі і rowspan – для вказівки числа об'єднуваних клітинок по вертикалі. Приклад об'єднання елементів таблиці приведені в лістингах 3.2 і 3.3.

**Лістинг 3.2 – Об'єднання клітинок по вертикалі**

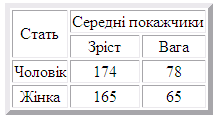
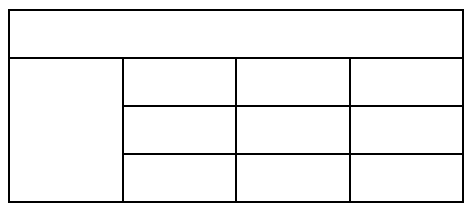


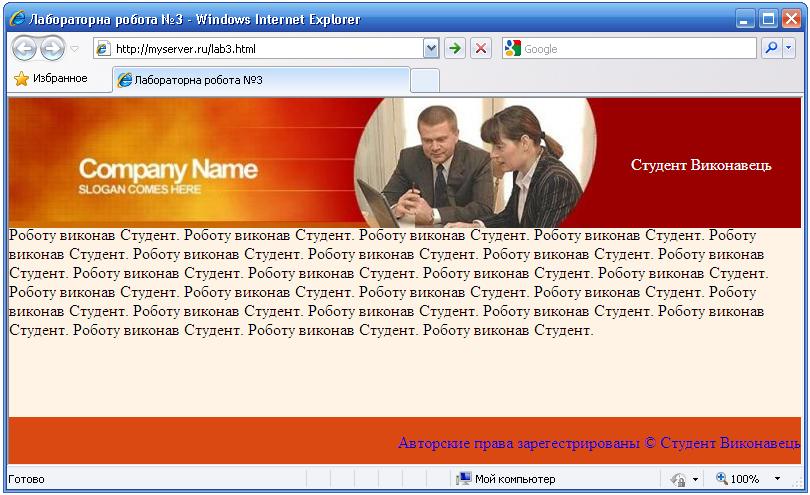
**Листинг 3.3 – Об'єднання клітинок по горизонталі**



Приклад створення таблиці для створення сайту, буде розглянутий в практичній роботі

**3.2 Практична частина**

1. Запустіть веб-сервер Денвер.
2. Створіть файл **lab3.htm** і збережіть його в своїй папці на веб-сервері.
3. Створіть у файлі **lab3.htm** таблицю розміром 3х3 для гри в «хрестики-нулі». Для створення таблиці використовують тег **<table>.** Для таблиці необхідно створити рамку для відображення меж. Порожні клітинки заповніть символом нерозривного пробілу. 
4. Доповнити файл таблицею з об'єднаними клітинками і жирнішою рамкою для всієї таблиці, текст вирівняти по середині клітинки (вертикально і горизонтально).
5. Створити таблицю, встановити їй наступні параметри: розташування на екрані – по центру, ширина 70% від ширини екрану. Колір вертикальної спільної клітинки – блакитний, спільної горизонтальної клітинки – жовтий. Колір останньої частини таблиці – зелений. Вирівнювання тексту у клітинках – по правому краю.
6. Створити в окремому файлі таблицю для шаблону сайту, яка відповідатиме наступним вимогам (див. рис.3.1).

****

**Рис.3.1. Шаблон сайту**

* шапка таблиці повинна складатися з двох клітинок;
* розмір лівої верхньої клітинки задати залежно від розміру зображення;
* фоном цієї клітинки зробити зображення (атрибут **background** для тега клітинки **<td>**);
* права верхня клітинка повинна займати місце, що залишилося у вікні браузеру і мати колір фону #.... (дізнатися колір фону можна використовуючи кнопку **select** у вікні вибору кольору фону);
* для того щоб прибрати відстань між клітинками використовуйте атрибути **cellpadding** і **cellspacing** тега **<body**>;
* у верхній правій клітинці змініть колір тексту і його вирівнювання (згідно рисунку) і запишіть в ньому своє прізвище. Для вирівнювання тексту необхідно використовувати атрибути **aling** і **valign**;
* центральну частину (ту, що містить основний текст) зробити заввишки в 350 пікселів і задати колір фону **#fff3e5**;
* у центральній клітинці об'єднати два клітинки так, щоб вийшла одна спільна, встановити вирівнювання тексту по ширині і по верхньому краю клітинки;
* задати підвал таблиці заввишки в 30 пікселів, текст вирівняти по правому краю. Включити в підпис символ **©**.

1. У окремому файлі створити таблицю розміром 3х3 (пунктиром вказані кордони клітинок).
2. Задати ширину і висоту таблиці і клітинок так, щоб ширина таблиці зменшувалася і збільшувалася при зміні розмірів вікна браузера, а висота була встановлена рівною висоті браузера. Якщо клітинка порожня, то вона не відображуватиметься на екрані, а якщо у клітинці знаходиться пробіл, то її висота може бути більше необхідної, для зменшення висоти вічка слід використовувати замість нерозривного пробілу прозору картинку розміром в 1 піксель (картинка є в архіві до лабораторної роботи).
3. Додайте у файл **index.htm** посилання на виконані завдання з лабораторної роботи №3.
4. Покажіть виконану роботу викладачеві.

**3.3 Контрольні питання**

1. Який тег служить для вставки таблиці в html-документ?
2. Які атрибути тега **<table>** дозволяють змінити фон таблиці і ширину межі таблиці?
3. Чи задається ширина межі таблиці одна для всієї таблиці або задається окремо для зовнішньої межі і внутрішніх комірок?
4. Який тег відповідає за початок рядка?
5. Який тег відповідає за початок клітинки?
6. Як змінити вирівнювання тексту у клітинці?
7. Чи може бути в одному рядку клітинки з різним вертикальним вирівнюванням?
8. Якщо для таблиці встановити колір фону і фонове зображення одночасно, що відображуватиметься на екрані?
9. Як можна об'єднати клітинки по горизонталі?
10. Як можна в таблиці об'єднати клітинки по вертикалі?

**4 ЛАБОРАТОРНА РОБОТА №4**

| **Тема:** | **Форми** |
| --- | --- |
| **Мета:** | 1. Навчитися створювати форми і вказувати обробників форм; 2. Навчитися створювати елементи керування форми; 3. Закріпити навики, отримані при створенні таблиць. |
| **Час:** | 2 год. |

**4.1 Теоретичні відомості**

4.1.1 Форми. Загальні відомості

Форми призначені для організації взаємодії з користувачем. Вони дозволяють вводити текст, здійснювати вибір із запропонованих значень за допомогою списків або кнопок, організовувати інтерактивний обмін інформацією між Web-сторінкою і сервером.

Як правило, форма працює спільно зі встановленим на сервері сценарним забезпеченням, що оброблятиме введену інформацію. Механізм обробки форм представлений на рис.4.1.



**Рис.4.1. Обробка даних форми сервером**

Найширше форми застосовуються для пошуку, реєстрації, заповнення анкет, тестів та ін. Форма визначається за допомогою тегів **<form>...</form>,** між якими розташовуються поля введення, кнопки, а також всі необхідні елементи оформлення форми. Для вказівки параметрів в тезі **<form>** використовуються атрибути (див. таблицю 4.1).

Таблиця 4.1

**Атрибути тега <form>**

| **Атрибут** | **Призначення** |
| --- | --- |
| 1 | 2 |
| action | адреса програми або документа, які обробляють дані форми |
| method | Можливі варіанти значень методу: *GET* – спосіб передачі, коли дані, введені у форму додаються в рядок запиту та *POST* – дані пердаються у прихованому вигляді |

Продовження таблиці 4.1

| name | ім'я форми. Використовується для звернення до форми за допомогою javscript |
| --- | --- |
| target | ім'я вікна або фрейма, куди обробник завантажуватиме повернений результат |
| enctype | тип інформації форми. Розрізняється в разі використання символьно-цифровий інформації і в разі використання файлів |

4.1.1 Елементи керування форми

Сама форма служить лише обов'язковим контейнером для розміщення елементів керування. Зовнішній вигляд елементів керування залежить від встановлених значень атрибутів.

**Поле введення** 

Однорядкове поле введення використовується, коли необхідно, щоб користувач ввів у форму дані в довільній формі але обмежені за об'ємом. Для отримання поля введення використовують тег **<input** **type=“text”>.**

Тег **<input>** є одним з різносторонніх елементів форми і дозволяє створювати різні елементи інтерфейсу і забезпечити взаємодію з користувачем. Головним чином **<input>** призначений для створення текстових полів, різних кнопок, перемикачів і прапорців.

Основний параметр тега **<input>,** що визначає вигляд елементу, – **type**. Він дозволяє задавати наступні елементи форми: текстове поле (**text**), поле з паролем (**password**), перемикач (**radio**), прапорець (**checkbox**), приховане поле (**hidden**), кнопка (**button**), кнопка для відправки форми (**submit**), кнопка для очищення форми (**reset**), поле для відправки файлу (**file**) і кнопка із зображенням (**image**). Для кожного елементу існує свій список параметрів, які визначають його вигляд і характеристики.

Окрім атрибуту **type** тег **<input>** має наступні атрибути (див. таблиця 4.2):

Таблиця 4.2

**Атрибути тега <input>**

| **Атрибут** | **Призначення** |
| --- | --- |
| 1 | 2 |
| align | визначає вирівнювання, у випадку якщо **type=“image”** |
| alt | визначає альтернативний текст для кнопки із зображенням (**type=“image”**) |
| checked | заздалегідь активований перемикач або прапорець |
| disabled | блокує доступ і зміну елементу (поле недоступне) |
| maxlength | максимальна кількість символів, дозволених в тексті |

Продовження таблиці 4.2

| name | ім'я поля, призначене для того, щоб відрізняти один елемент керування від іншого. Ім'я може бути вказане по всім правилам іменування змінних |
| --- | --- |
| readonly | дозволяє лише читання з текстового елементу керування |
| size | ширина текстового поля в символах |
| value | значення елементу. Значення змінної вказаної в атрибуті **name** |

**Перемикач **

Перемикачі визначають поля вибору одного значення з декількох доступних, для кожної позиції перемикача створюється свій тег **<input** **type=“radio”>.** Групуються перемикачі за допомогою однакового імені, що задається атрибутом **name.**

**Прапорець **

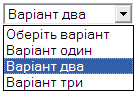
Прапорці використовуються, коли необхідно, щоб користувач вибрав один або декілька варіантів з обмеженого числа варіантів вибору. Прапорці у формі не залежать один від одного, їх можна встановити або скинути в будь-якій комбінації. Для кожного прапорця необхідно задати своє унікальне ім'я за допомогою атрибуту **name.** Створюються прапорці тегом **<input type=“checkbox”>.** Для встановки прапорця при завантаженні сторінки необхідно вказати атрибут **checked=“checked”.**

**Командна кнопка** 

Командна кнопка відправки **(type=“submit”)** використовується для виконання пересилки даних форми на сервер. Командна кнопка скидання **(type=“reset”)** повертає форму до вихідного достатку (очищає форму).

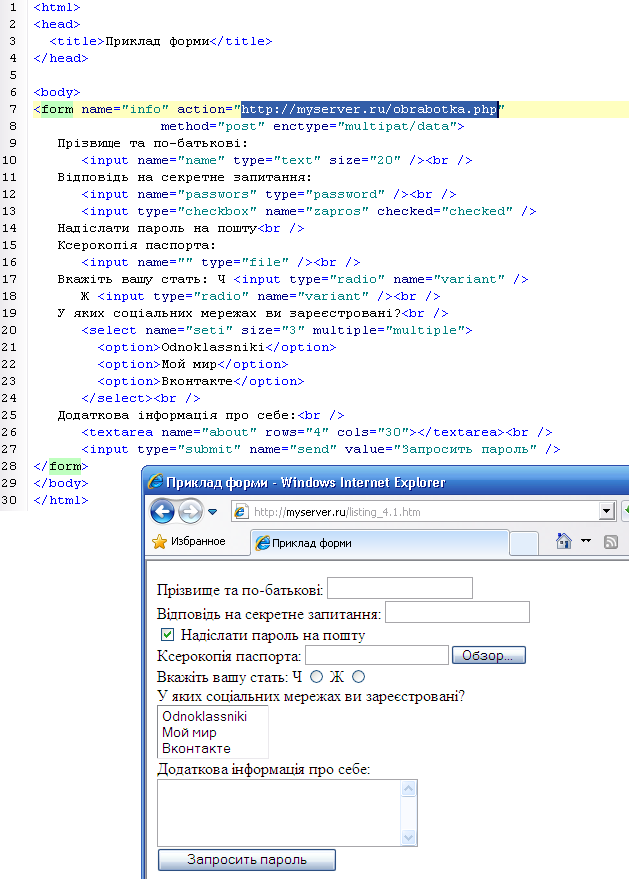
**Поле вибору файлу **

Поле вибору файлу **(type=“file”)** створює поле для вибору файлу, який буде завантажений на сервер разом з інформацією форми. Поряд з полем введення відображується командна кнопка “Огляд...”, що відкриває стандартне діалогове вікно вибору файлу. Якщо форма має на увазі завантаження файлів на сервер, то в атрибутах форми мають бути встановлені наступні атрибути **method=“post”** і **enctype=“multipart/form-data”.**

**Списки вибору** 

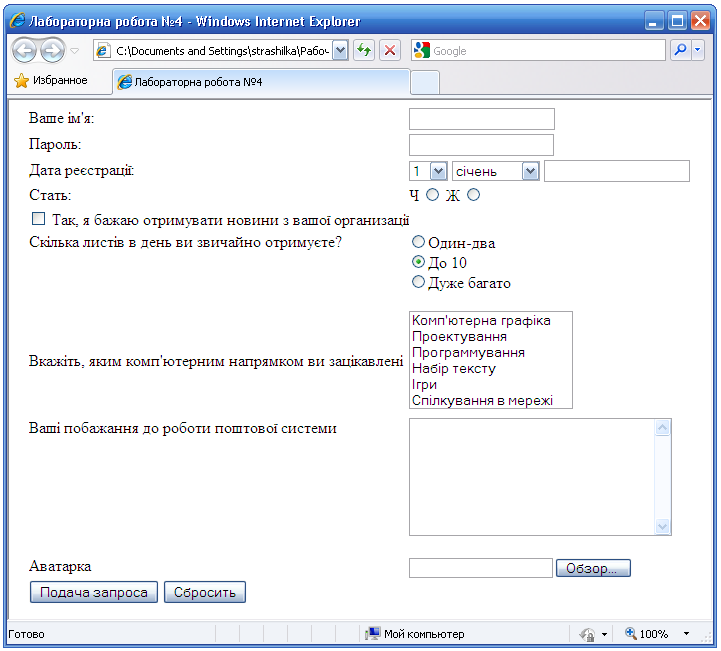
Списки вибору бувають двох типів: списки, що розкриваються (випадні меню), і списки з множинним вибором. Незалежно від типів списків описуються вони однаково за допомогою пари тегів **<select> </select>.** Окремі елементи списку задаються з використанням тега **<option>.** Тип списку визначається за допомогою атрибуту **multiple** тега **<select>.**Приклад використання форми наведений в лістингу 4.1:

**Лістинг 4.3 – Використання форми і елементів керування**



**4.2 Практична частина**

1. Запустіть веб-сервер Денвер.
2. Створити форму для заповнення анкети (див. архів до лабораторної роботи). Зверніть увагу що сторінка з анкетою зверстана у вигляді таблиці з невидимою рамкою (див. рис.4.2).
3. Збережіть анкету у файлі **anketa.htm** на своєму домені. Зверніть увагу, що вся форма повинна розташовуватися у тезі **<form>**.



**Рис.4.2. Форма заповнення анкети**

1. Створити поля для введення імені користувача і пароля користувача (символи вводяться в поле Пароль повинні відображуватися зірочками). Вставка полів здійснюється тегом **<input>** з різними значеннями атрибуту **type**.
2. Створити поля для введення дати народження. День і місяць народження вибирається з випадного списку, а рік вводиться в ручну в полі введення, причому кількість символів, що вводяться, не повинна перевищувати чотири. Для формування списку використовувати тег **<select>,** а для завдання обмеження на кількість символів, що вводяться, атрибут **maxlength** тега **<input>.**
3. Створити перемикачі для вибору статі. Реалізувати можливість вибору лише одного варіанту. Використовувати тег **<input>.**
4. Створити прапорець, з текстом «Так, я бажаю отримувати спам» з прапорцем, встановленим за умовчанням. Використовувати тег **<input>.**
5. Створити перемикачі для вибору кількості листів отримуваних в день, перевірити щоб перемикачі із завдання 6 і 8 формували дві різні групи.
6. Створити список інтересів з можливістю вибору декількох елементів.
7. Додати поле для введення декількох рядків тексту з додатковою інформацією про користувача. Поле повинне мати розміри 7 рядків і 30 стовпців. Для вставки такого поля використовуйте тег **<textarea>**.
8. Створити поле з кнопкою «Огляд» для завантаження файлу зображення. Використовувати тег **<input>**.
9. Створити дві кнопки. Першу для передачі даних форми на сервер, а другу для очищення форми. Використовувати тег **<input>**.
10. Покажіть виконану роботу викладачеві.

**4.3 Контрольні питання**

1. У якому тезі розташовані всі елементи керування?
2. Який атрибут тега **<form>** вказує на файл, який оброблятиме дані форми?
3. Які існують значення атрибуту **type** в тезі **<input>**?
4. У чому відзнаки тега **<input type=“reset”>** і **<input type=“submit”>**?
5. Як додати на форму перемикач?
6. Для чого призначений тег **<textarea>**?
7. Скільки тегів необхідно щоб вивести на екран поле введення для вказівки шляху до файлу і кнопки огляд?
8. Які існують методи передачі даних форми на сервер?
9. Як здійснюється обробка форм?
10. За рахунок чого перемикачі можна розбивати на групи?
11. Який атрибут тега **<input>** дозволяє вказати значення, яке буде заповнено при першому відображенні форми?
12. Для чого призначений атрибут **checked**?

**5 ЛАБОРАТОРНА РОБОТА №5**

| **Тема:** | **Фрейми. Карти зображень** |
| --- | --- |
| **Мета:** | 1. Навчитися створювати фрейми; 2. Навчитися управляти властивостями фреймів; 3. Навчитися створювати і застосовувати карти зображень для непрямокутних посилань. |
| **Час:** | 2 год. |

**5.1 Теоретичні відомості**

5.1.1 Фрейми

Фрейми дозволяють розбивати веб-сторінки на скроллюємі підвікна з метою поліпшення зовнішнього вигляду і функціональності інформаційних систем і веб-додатків. Кожен фрейм має наступні властивості:

– фрейм має свій URL, що дозволяє завантажувати його незалежно від інших фреймів;

– фрейм має власне ім'я (атрибут **name**), що дозволяє переходити до нього з іншого фрейма (атрибут **target** в тезі посилання **<a>**);

– розмір фрейма може бути змінений користувачем прямо на екрані за допомогою миші (якщо це не заборонено вказівкою спеціального параметра).

Ці властивості фреймів дозволяють створювати інтерфейсні рішення, які можуть поєднувати статичну інформацію в одному фреймі (це може бути зміст, графічний логотип фірми або набір кнопок, що управляють) і динамічну інформацію в іншому фреймі. Таким чином в одному фреймі знаходиться власне запит, а в іншому результати запиту.

Формат документа, використовуючого фрейми, зовні дуже нагадує формат звичайного документа, лише замість тега **<body>** використовується контейнер **<frameset>,** що містить опис внутрішніх html-документов, містить власне інформацію, що розміщується у фреймах. Контейнер **<frameset>,** у свою чергу складається з самих фреймів – тег **<frame>** і **<noframe>** дозволяє будувати подвійні документи для браузерів, підтримуючих фрейми і не підтримуючих фрейми.

Тег **<frameset>** має два взаємовиключні атрибути: **rows** (визначає число і висоту рядків в наборі фреймів) і **cols** (визначає число і ширину стовпців в наборі фреймів). Ширина (або висота) фрейма може задаватися як в пікселях, так і у відсотках від спільної ширини (або висоти) вікна браузера.

Наприклад:

**<frameset cols=”50,\*,50“> –** описує три фрейма, два по 50 пікселів справа і зліва, та один усередині цих смужок.

**<frameset rows=”20%,3\*,\*“> –** описує три фрейма, перший з яких займає 20% площі зверху екрану, другий 3/4 що залишився від першого фрейма (тобто 60% всієї площі вікна), а останній 1/4 (тобто 20% всієї площі вікна).

Тег **<frame>** має свої атрибути, які детально описують вміст і поведінку кожного фрейма (див. таблиця 5.1):

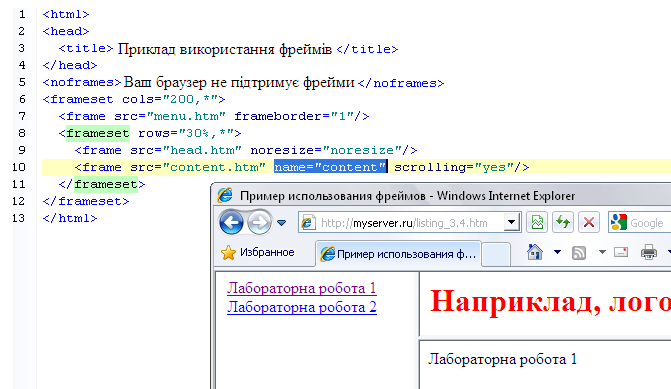
Таблиця 5.1

**Атрибути тега <frame>**

| **Атрибут** | **Призначення** |
| --- | --- |
| src | задає шлях до файлу, призначеного для завантаження у фрейм |
| name | задає унікальне ім'я фрейма |
| scrolling | спосіб відображення смуги прокрутки у фреймі. Можливі варіанти значень**: yes, no, auto** |
| bordercolor | колір лінії межі |
| frameborder | визначає чи слід відображати рамку довкола фрейма |
| norsize | визначає чи можна змінювати розмір фрейма користувачеві чи ні |

Приклад використання фреймів розглянутий в лістингу 5.1:

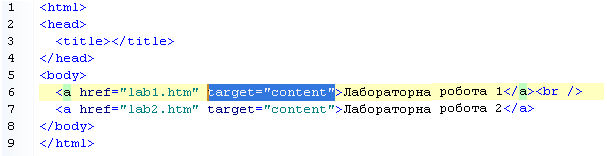
**Лістинг 5.1 – Приклад використання фреймів**



*Коментарій до лістингу.*

1. в документі відсутній тег **<body>;**
2. спочатку фреймова структура розбита на два стовпці: перший шириною в 200 пікселів, а другою займає простір, що залишився;
3. у лівому фреймі відкритий файл menu.htm і посилання цьому файлу організовані з вказівкою цільового фрейма, в якому повинні відкривати нові файли;
4. замість правого фрейма організована нова фреймова структура з двох рядків. Висота першого рядка 30% від спільної висоти вікна, а нижня частина займає ті 70%, що залишилися;
5. у верхньому файлі відкритий файл header.htm;
6. у нижньому файлі відкритий файл content.htm або той, який буде вибраний в лівому фреймі. Саме на ім'я тега **content** (див. лістинг 5.1) і по атрибуту **target** в тезі **<a>** файлу menu.htm (див. лістинг 5.2) визначається де має бути відкрите посилання.

**Лістинг 5.2 – Атрибут target в тезі <a>**



5.1.2 Карти зображень

Карти-зображення дозволяють прив'язувати посилання до різних областей одного зображення. Реалізуються карти зображень в двох різних варіантах – серверному і клієнтському.

В разі використання серверного варіанту браузер посилає запит на сервер з вказівкою координат курсора миші над зображенням для здобуття адреси вибраного посилання і чекає відповіді з необхідною інформацією. Такий підхід вимагає додаткового часу на чекання результату і окремі файли для кожної карти-зображення.

У клієнтському варіанті карта розташовується в тому ж HTML-документі, що і посилання на зображення. Клієнтський варіант є більш частіше використовуваним. Для вказівки браузеру, що зображення є картою, використовується атрибут **usemap** в тезі **<img>.** Він є посиланням на опис конфігурації карти, яка задається тегом **<map>.** Значення атрибуту **name** даного тега повинне відповідати імені в **usemap**. Для завдання активної області, що є посиланням на HTML-документ, використовується тег **<area>** з атрибутами (див. таблицю 5.2):

Таблиця 5.2

**Атрибути тега <area>**

| **Атрибут** | **Призначення** |
| --- | --- |
| shape | визначає форму активної області. Може набувати значень **circle**, **rect**, **poly** |
| alt | додає альтернативний текст підпису посилання для кожної області |
| cords | задає координати активної області. Координати відлічуються в пікселях від лівого верхнього кута зображення. Перше число є координатою по горизонталі, друге – по вертикалі. |

Список координат переданих в атрибут **coords** залежить від форми області.

Для кола задаються три числа – координати центру кола і радіус: **<area shape="circle" coords="230,340, 100" href="circle.html">**

Для прямокутника – координати лівого верхнього і правого нижнього кута: **<area shape="rect" coords="24,18, 210,56" href="rect.html">**

Для полігону задаються координати його вершин.

В разі використання посилань непрямокутної форми альтернативою картам зображення може бути впровадження flash.

**5.2 Практична частина**

1. Запустіть веб-сервер Денвер.
2. Створіть html-файл (frame.htm), що складається з трьох фреймів (див. рисунок справа). Для створення фрейма необхідно після розділу заголовка документа вказати структуру фреймів: два фрейми вертикальні, а в другому вертикальному ще два горизонтальні фрейми.

**<frameset cols="20 \*">**

**<frame src="file.htm">**

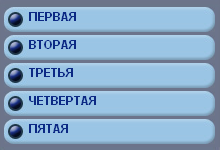
**<frameset rows="100 \*">**

**<frame src="file.htm">**

**<frame src="file.htm">**

**</frameset>**

**</frameset>**

1. Розташуйте файл на своєму веб-сервер і дайте кожному фрейму унікальне ім'я: значення атрибуту **name** для тега **<frame>.**
2. Задайте лівому фрейму фіксовану ширину в 234 пікселі із забороною змінювати ширину фрейма (атрибут **noresize**).
3. У лівий фрейм помістіть файл меню (**menu.htm**), який складатиметься із списку лабораторних робіт. Меню розмістіть в таблиці, що складається з трьох рядків (у кожен рядок фоном помістіть рисунок).
4. Задайте для кожної лабораторної роботи посилання на відповідний документ, який буде відкритий в правому нижньому фреймі. Для відкриття посилання в іншому фреймі використовуйте атрибут **target** тега **<a>,** як значення атрибуту **target** вкажіть ім'я фрейма в якому необхідно відкрити файл.
5. У верхній фрейм помістіть, пошукову систему google.com. Значення атрибуту **src** для другого тега **<frame>.**
6. Створіть в папці свого домену файл **map.html**, що містить карту зображень.
7. Помістіть у файл map.html зображення-карту з геометричними фігурами:

<img src="figure\_1.png" border="0" usemap="#navigation">

1. Доповніть файл обробкою карти зображень

<map name="navigation">

<area shape="circle" coords="72,93,40"

href="krug.html" title="Коло">

<area shape="rect" coords="129,129,268,194"

href="pryam.html" title="Прямокутник">

<area shape="poly" coords="168,108,221,15,279,108"

href="triangle.html" title="Трикутник">

</map>

1. Перевірте роботу карти зображень. Зверніть увагу на спливаючі підказки і посилання.
2. Створіть в окремому файлі аналогічну карту зображень для завдання областей України. У карті зображень виділіть три області, на київську область призначте посилання на сайт **meta.ua** на дві інші області будь-які посилання. Щоб узнати координати точок зображення можна використовувати програму Paint .
3. Додайте у файл, що знаходиться в лівому фреймі посилання на обидві карти зображень.
4. Покажіть виконану роботу викладачеві.

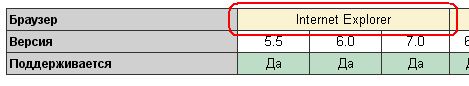
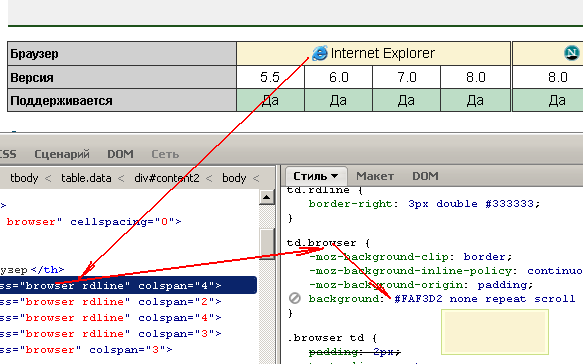
**5.3 Контрольні питання**

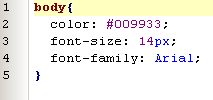
1. Для чого використовуються фрейми?
2. За що відповідає атрибут **target** тега **<a>**?
3. Як вказати посиланню в якому фреймі треба відкритися?
4. У яких одиницях можна задати ширину або висоту фрейма?
5. Чи можна змінювати межі між фреймами?
6. Для чого використовуються карти зображень?
7. Які форми фігур можуть використовуватися в картах зображень?
8. Як можна підписати фігури карти зображень?
9. Які види карт зображень бувають?
10. Що може служити альтернативою використання карт зображень?

**ЛАБОРАТОРНА РОБОТА №6**

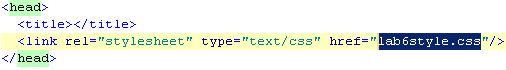
| **Тема:** | **Введення в каскадні таблиці стилів (CSS).** |
| --- | --- |
| **Мета:** | 1. Вивчити можливості доповнення **Firebug** до програми FireFox по перегляду структури HTML-документа; 2. Вивчити підключення таблиць стилів до документа; 3. Навчитися змінювати css-файли з використанням доповнення **FireBug**. |
| **Час:** | 2 год. |

**6.1 Практична частина**

1. Запустіть програму **FireFox**.
2. Перевірте, що доповнення **Firebug** не відключене. Для цього необхідно зняти прапорець із пунктів **Disable...** в меню **Инструменты (Firebug**. Доповнення **Firebug** дозволяє досліджувати властивості будь-якого фрагмента html-документа з урахуванням стилів.
3. Перегляньте структуру html-документа. Для цього необхідно вивести панель доповнення **Firebug(Инструменты(Firebug(Open Firebug)**. В нижній частині вікна з'явиться панель **Firebug**, в якій необхідно натиснути кнопку **Inspect** і підвести до будь-якого елемента документа, при цьому в нижній частині вікна буде відображена структура документа. Зверніть увагу на деревоподібну структуру web-сторінки та підключення стилів в заголовку документа. 
4. Із включеною кнопкою **Inspect** зафіксуйте покажчик миші на жовтому осередку. У вікні доповнення **Firebug** ліворуч буде відображена структура html-документа. У вікні праворуч будуть відображені стилі з підключеного файлу стилів. Змінюючи значення властивостей стилів, можна одержувати зовсім нові зображення.
5. Подивіться який клас призначений виділеному осередку.
6. Знайдіть у правій колонці клас виділеного осередку та змініть властивість тла (backgroud), наприклад на значення #FF6600 жовтогарячих кольорів. Зверніть увагу, що якщо ви не коректно вказали властивість або його значення, то визначення зникне з вікна стилів.
7. Змініть кольори тла всіх зелених осередків на голубий. Таким чином, використовуючи **FireBug** можна переглядати та змінювати стилі в документі. Зберігати результати змін, на жаль, не можна.
8. Запустіть HTMLPad і створіть в ньому два файли: **lab.html** та **lab6style.css**. Обидва файли збережіть на своєму домені.
9. Створіть в html файлі п'ять абзаців довжиною в три рядки.
10. Збережіть html сторінку та прогляніть її на своєму домені.
11. В css файлі напишіть стиль для селектору тегу body, в якому вкажіть кольори тексту, розмір букв і назву шрифту:



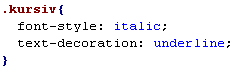
1. Включіть попередній перегляд і подивіться як виглядає стил усього документа. Для того, щоб html документ «підняв» ці стилі необхідно в html-файлі підключити css файл:

**

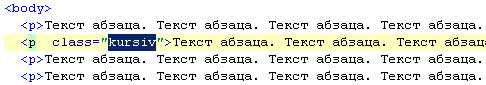
1. Збережіть зміни. Подивіться html-файл у браузері. Зверніть увагу на шрифт.
2. В css файлі створіть шрифт абзацу, у якому встановіть наступний інтервал тексту між буквами:

**

1. Збережіть css файл. Оновіть html файл у браузері. Використовуючи доповнення **FireBug** перегляньте які стилі використовуються в другому абзаці.
2. Для того, щоб один з абзаців зробити курсивом та підкреслити, необхідно створити наступний клас

**

17. Підключіть даний клас до другого абзацу



18. Збережіть зміни та перевірте результат.

19. Створіть клас для форматування третього абзацу з наступними властивостями:

* шрифт абзацу Comic Sans; (**font-family**)
* розмір шрифту 25 пт; (**font-size**)
* відступ нового рядка в 50 пикселів (**text-indent**).

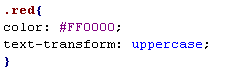
20. Підключіть стиль до третього абзацу (використовуючи атрибут **class**).

21. Створіть класс **.rich** для відображення п'ятого абзацу, що містить наступні властивості тексту:

* Шрифт Arial, 15 пт;
* Накреслення жирне + курсив (**font-weight; font-style**);
* Колір букв жовтогарячий (**color**);
* Висота рядка 25 пунктів (**line-height**).

22. Створіть клас **perenos** і забороніть в ньому перенесення слів в межах одного абзацу (**white-space**). Призначте цю властивість другому та четвертому абзацу. Для цього в атрибуті **class** тегу абзацу через пробіл вкажіть два класи (той що був раніше + клас **perenos**). Збережіть зміни.

23. Для того що б виділити слова в межах одного абзацу можна використати тег **<span>** (він дозволяє додавати форматування до будь-якого текстового елемента). Створюємо клас, що містить заголовні червоні букви:

і вказуємо новий клас у тегу **<span>** 

**

24. Створіть клас з наступними властивостями:

* Шрифт Courier 12 пт;
* всі букви заголовні;
* міжбуквений інтервал виряджений на 200%;
* текст підкреслений;
* міжрядковий інтервал подвійний.

25. Додайте ще пару абзаців і призначте їм властивості цього класу.

26. Створіть свої 2-3 класи стилів тексту, які ви будете використовувати в практичній роботі (використовуючи властивості таблиці 6.1). Кожен стиль повинен включати по 3-4 властивості форматування тексту і бути підключеним до абзаців або до заголовків першого, другого та третього рівнів.

Таблиця 6.1

**Властивості тексту**

| **№** | **Властивість (можливі значення)** | **Призначення властивості** |
| --- | --- | --- |
|  | font-family (Arial | Times New Roman | Courier | Tahoma) | назва шрифту |
|  | font-weight (normal | bold | lighter) | насиченість шрифту (ширина букв) |
|  | font-size (small | large | medium | 120% | 14px) | розмір шрифту |
|  | font-style (normal | italic) | стиль шрифту (звичайно курсив) |
|  | font-stretch (normal | width | condensed | expanded) | розтягнуте накреслення шрифту |
|  | text-indent (число px | %) | відступ першого рядка, при негативному значенні виступ |
|  | text-align (left | center | right | justify) | горизонтальне вирівнювання |
|  | line-height (normal | …px | …% | …em) | висота рядка (міжрядковий інтервал) |
|  | vertical-align (sub | sup | top | middle | bottom) | вертикальне вирівнювання тексту в контейнері |
|  | word-spacing (…px|normal) | відстань між словами |
|  | letter-spacing (…px|normal) | відстань між буквами |
|  | text-transform (uppercase | lowercase | capitalize | none) | регістр букв |
|  | text-decoration (none | underline | overline | line-through) | оформлення тексту |
|  | text-shadow (кольори довжин довжина довжина) | тінь до тексту |
|  | white-space (normal | nowrap | pre) | обробка пробілів |
|  | color | кольори |

**6.2 Контрольні питання**

1. Для чого використовують доповнення **Firebug**?
2. Наведіть синтаксис правила в CSS?
3. Назвіть способи застосування таблиці стилів до HTML документа?
4. Наведіть властивості тексту?
5. Наведіть властивості шрифту?
6. Наведіть властивості кольору і тла?

**ЛАБОРАТОРНА РОБОТА №7**

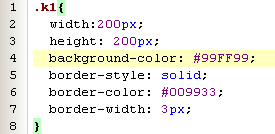
| **Тема:** | **Властивості блокових об'єктів.** |
| --- | --- |
| **Мета:** | 1. Вивчити можливості CSS по роботі з блоковими об'єктами; 2. Вивчити основні властивості блоків; 3. Навчитися застосовувати теги **<div>** та **<span>.** |
| **Час:** | 2 год. |

**7.1 Практична частина**

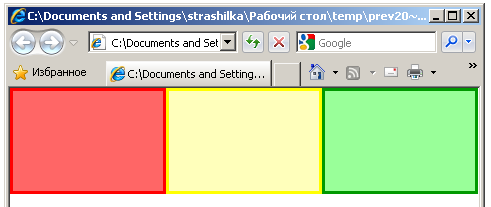
1. Створіть html-файл lab7.htm, що містить блочний елемент DIV.

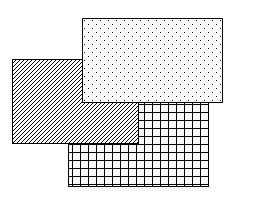


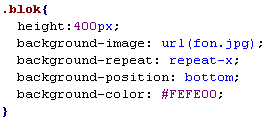
1. Створіть css-файл і підключіть його до html-документа.
2. Створіть клас k1, в якому визначте розмір блоку 200х200 пикселів, тло (ясно-зелений колір) і рамку (суцільна, темно-зелена, шириною в 3 пикселя).

**

1. Підключіть клас k1 до блоку в html-документі. Зверніть увагу на те, як зміниться блок.
2. Додайте ще п'ять таких самих блоків розміром 200х200 пикселів. Блоки розташовуються один під одним.
3. Для того, щоб блоки розташовувалися один за одним по горизонталі необхідно додати в клас k1 оголошення . Оновіть і подивиться на зміни при зміні розмірів вікна.
4. Тепер блоки «липнуть» один до одного, між ними немає відстані, для того щоб додати зовнішній відступ ліворуч додаємо оголошення . Збережіть, перегляньте зміни. У браузері Mozilla виділіть будь-який **div** (за допомогою **FireBug**) і перейдіть на вкладку **Layout**. Поексперементуйте із властивостями: **padding, margin, border** та **offset**. Зверніть увагу, що при зміні всіх цих властивостей, вони дописуються в html-код.
5. Створіть в окремому файлі горизонтальний світлофор із трьох блоків. Ширину блоків зробіть динамічної, що б вони змінювалася залежно від ширини вікна, але всі блоки повинні мати одну ширину. Задайте кожному блоку свої кольори. Відступи для **body** заберіть за допомогою css.

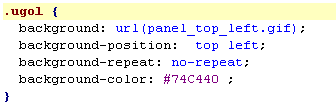
**

1. Задайте першому блоку оголошення **display:none**, а іншому **visibility: hidden;** В чому відмінності цих двох оголошень?
2. Створіть файл, а в ньому три блоки. У першому блоці зробіть рамку з різним типом лінії для кожної сторони блоку. В другому блоці задайте фонове зображення (**background-image**). В третьому блоці, задайте тло…Додавайте рядки по одному та дивіться на зміни в браузері після кожного рядка.

**

1. Розташуйте блоки один над одним. Для цього встановіть кожному блоку оголошення **position: absolute;** і порядок розташування блоків зверху донизу: оголошення **z-index: 5;** (у тих блоків що нижче, значення індексу повинне бути меншим, наприклад 10, 20 та 30). Для того, щоб блоки починалися не в одній точці, їм необхідно задати властивості **left** і **top**.
2. Створимо округлені кути. Для цього створюємо чотири вкладених один в одний блоки.

   
опишемо для зовнішнього блоку клас ugol і призначимо цей клас блоку



опишим клас для правого верхнього кута



і самостійно для лівого і правого нижнього кутів.

1. Створити новий html файл у який помістити зображення ягоди.
2. В css-файл підключений до html-файлу запишіть стиль для зміни прозорості будь-якого зображення при наведенні на нього курсору миші.



Для Internet Explorer рядки повинні бути наступними:



1. Використовуючи тільки блоки, створіть поле для гри в хрестики-нуліки.
2. Використовуючи блоки, їхню вкладеність, тло, вирівнювання, властивість **float**, шрифт **Georgia** створіть наступну шапку. Кути виріжте самостійно.**

**7.2 Контрольні питання**

* 1. Як розташувати блоки один під одним, та по горизонталі?
  2. Як розташувати блоки один над одним?
  3. Яким чином регулюється порядок розташування блоків зверху донизу?
  4. Пропишіть стиль для зміни прозорості будь-якого зображення при наведенні на нього курсору миші?

**ЛАБОРАТОРНА РОБОТА №8**

| **Тема:** | **Додаткові властивості CSS** |
| --- | --- |
| **Мета:** | 1) Детально вивчити структуру таблиці для прискорення створення стилів CSS;  2) Вивчити основні можливості роботи зі списками за допомогою CSS. |
| **Час:** | 2 год. |

**8.1 Практична частина**

1. Створіть html- файл і вставте у всередину тега **<body>** текст з файлу **текст.txt**.
2. Прогляньте і проаналізуйте структуру таблиці. Зверніть увагу на те, які теги можна використати в таблиці.
3. Створіть css- файл в якому вкажіть те, що в таблиці та осередках (**table, td, th{.}**) буде використовуватися рамка червоного кольору, шириною в 1 піксель. Під'єднайте цей css- файл до html- файлу.
4. Зверніть увагу, що рамка в таблиці виходить подвійна. Що б уникнути подвійної рамки, необхідно додати новий стиль.

\*{

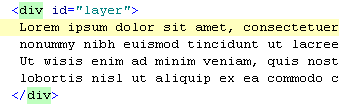
border-collapse: collapse;

}

1. Призначте таблиці окремий клас. Опишіть властивості окремих елементів таблиці використовуючи наступні стилі:

.frmtbl { /\*задание стиля таблицы в целом\*/ }  
.frmtbl thead { /\*задание стиля для заголовка таблицы \*/ }  
.frmtbl tbody { /\*задание стиля для основной части таблицы\*/ }  
.frmtbl tfoot { /\* задание стиля для нижней части таблицы \*/ }

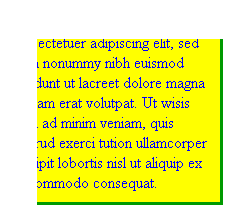
1. Використовуючи властивості **list - style - type**, **list - style - position**, **list - style - image** розмістіть у блоці маркірований список. В якості маркера використайте зображення, маркер має бути всередині списку. Відступ від лівого краю блоку до списку повинен складати 50 пікселів.
2. Створіть новий документ і розмістіть в ньому блок з ідентифікатором **layer.**

**

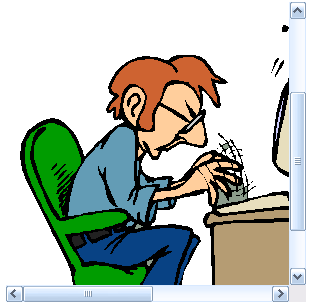
1. Опишіть ідентифікатор **layer**, який міститиме: жовтий фон, зелену рамочку, відступ від тексту до краю блоку до тексту, сині букви, ширину в 200 пікселів, абсолютне позиціонування і область позиціонування елементу.

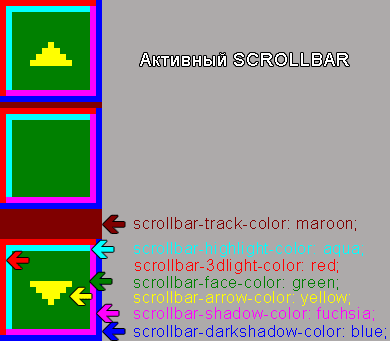
clip: rect(40px, auto, auto, 40px);

Якщо все зробили правильно, то на екрані повинна з'явитися тільки кадрована частина тексту.



1. Створіть блок з рамкою розміром 200х200 пікселів. Помістіть в цей блок зображення свідомо більшого розміру. Прогляньте в браузері. Поекспериментуйте з властивістю **overflow** таким чином, що у тому випадку, якщо у блок не вміщується вміст, з'являлися смуги прокрутки (чи тільки одна) :



10. В браузері Internet Explorer є можливість змінити колір смуг прокруток додавши наступні стилі. Наприклад:

html, body {

scrollbar-3dlight-color:red;

scrollbar-arrow-color:yellow;

scrollbar-highlight-color: aqua;

scrollbar-face-color:green;

scrollbar-shadow-color:fuchsia;

scrollbar-darkshadow-color:blue;

scrollbar-track-color: maroon;

}

**8.2 Контрольні питання**

1. Як призначити таблиці окремий клас, опишіть властивості окремих елементів таблиці?
2. Як отримати кадровану частину тексту?
3. Опишіть властивість **overflow**?

**ЛАБОРАТОРНА РОБОТА №9**

| **Тема:** | **Приклади верстання веб-сторінок** |
| --- | --- |
| **Мета:** | 1) Отримання практичних навичок верстання веб-сторінок. |
| **Час:** | 2 год. |

**9.1 Практична частина**

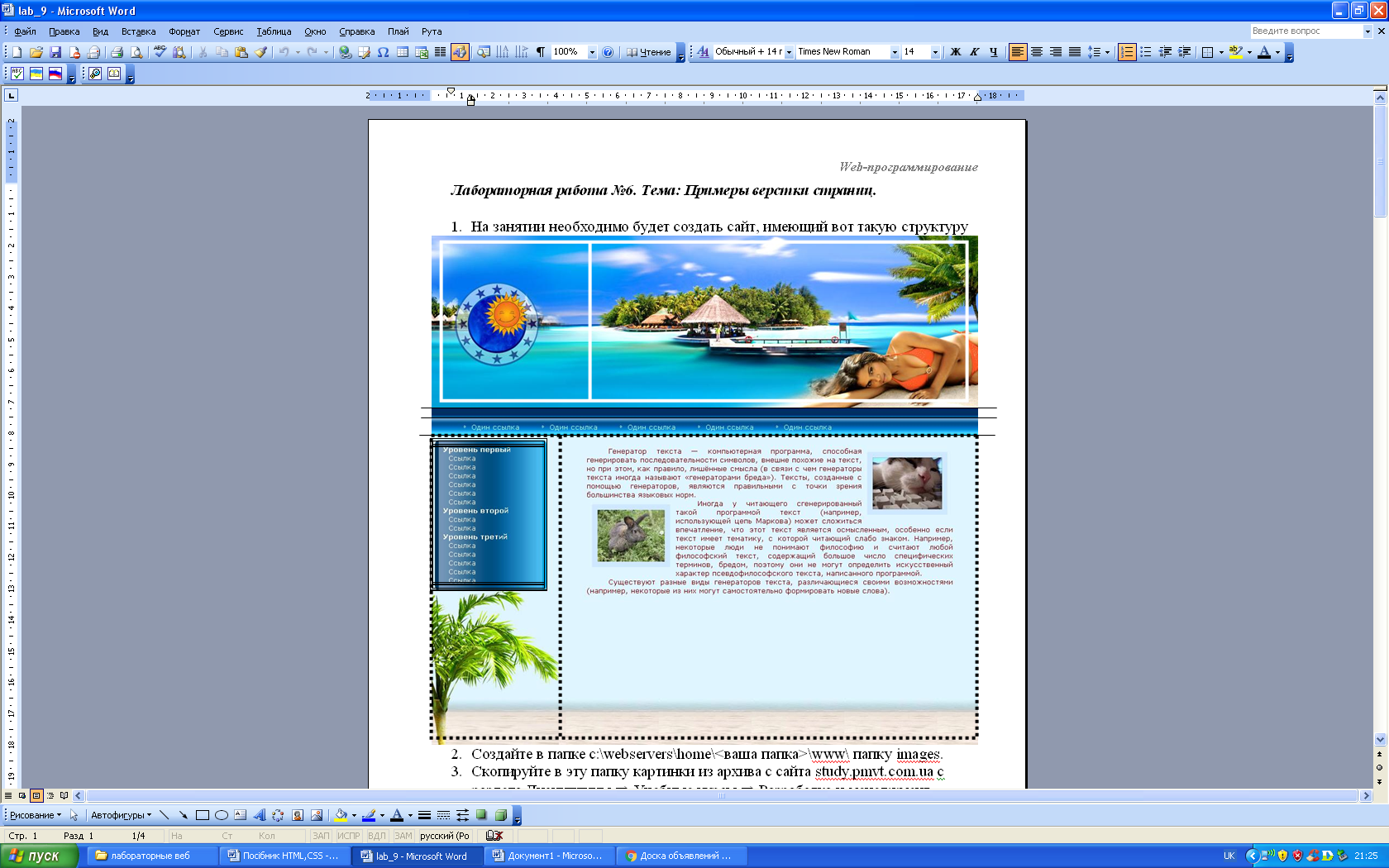
На даному лабораторному занятті необхідно створити веб-сайт, що має наступну структуру (див. рис.9.1). Для реалізації даного завдання необхідно виконати наступну послідовність дій.

1. Створіть в папці C:\Webservers\home\<ваша папка>\www\ папку images.
2. Скопіюйте в цю папку картинки з архіву до лабораторної роботи.
3. Створіть в програмі HTMLPad html-файл і збережіть його з ім’ям lab9.htm в папці C:\Webservers\home\<ваша папка>\www\.
4. Створіть в програмі HTMLPad css-файл і збережіть його з ім’ям styles\_7.css в папці C:\Webservers\home\<ваша папка>\www\.
5. Зв'яжіть html файл з css файлом. В тезі <head> пропишіть

<link type="text/css" rel="stylesheet" media="all" href="styles\_9.css">

1. Створіть заголовок web-сторінці "Лабораторная работа 9".

<title>Лабораторная работа 9</title>

Рис.9.1 – Приклад веб-сторінки

1. Опишіть в css файлі параметри всього html документу

body {

background-color:#DFF5FF;

font-size:13px;

margin:0;

}

1. Збережіть обидва файли (css і html) і прогляньте сторінку у браузері, вона має бути ніжно блакитного кольору і мати заголовок вікна.
2. Всередині тега <body> створіть таблицю з наступними параметрами: 1 стовпець, 4 рядки, ширина таблиці 100%, рамка 0 пікселів (скористайтеся для цього кнопкою таблиця на панелі інструментів) :

<table width="100%" cellpadding="0" cellspacing="0" border="0">

…

</table>

1. Перед початком кожного рядка додайте коментар з інформацією про номер рядка

<!--Первая строка-->.

***Перший рядок головної таблиці призначений для шапки***

1. Всередині першого рядка (всередині тега <td>) розташуйте таблицю, що складається з одного рядка і двох комірок всередині даного рядка.

<table width="100%" cellspacing="0" cellpadding="0" border="0">

…

</table>

1. В css файлі створіть клас для лівої комірки (файл css) і призначте його першій комірці <td class="top\_left"> (файл html)

.top\_left{

text-align: left;

vertical-align: top;

background-image: url(images/bg\_top.gif);

height: 318px;

}

1. В css файлі створіть клас для правої комірки і призначте його першій комірці <td class="top\_right">

.top\_right{

width: 713px;

background-image: url(images/big\_picture.jpg);

}

1. Розташуйте в першій (лівій) комірці логотип і посилання на ваш сайт (для цього використайте вставку зображення, а посилання в ньому можна буде вказати)

<a href="http://my.net">

<img src="images/logo.jpg" width="277" height="318" alt="" /></a>

1. Для того, щоб посилання не мали синьої контурної лінії добавте в зображення (тег <img>) атрибут border="0".
2. Подивіться на результат (верхня частина сторінки повинна цілком відповідати рис.9.1).

***Другий рядок головної таблиці (призначений для розділення шапки і меню)***

1. Для комірки другого рядка головної таблиці створіть і підключіть клас separator.
2. Клас повинен вміщувати наступні властивості: висота 19 пікселів, ширина 100 відсотків, фон зображення su\_bg.gif.
3. Збережіть, оновіть, подивіться на результат.

***Третій рядок головної таблиці призначений для верхнього горизонтального меню***

1. Створіть для комірки з рядком меню клас menu\_line з наступними параметрами: висота 32 пікселя, внутрішній відступ 60 пікселів, фонове зображення menu\_bg.gif.
2. Призначте комірці третього рядка класс menu\_line.
3. Створіть всередині комірки третього рядка блоковий об’єкт, в якій розташуйте зображення і посилання (створіть 5-6 копій блоків)

<div class="menu\_item">

<a href="#">Один ссылка</a>

</div>

1. Опишіть клас menu\_item в якому необхідно врахувати:

зображення тла (трикутник menu\_list.gif), скасувати повторення зображення тла, вирівняти тло по левому краю, внутрішній відступ ліворуч 15 пікселів, а зовнішній відступ праворуч 40 пікселів, блоки вкладати в ряд таким чином, щоб вони своїм лівим боком приєднувалися до попереднього блоку.

1. Збережіть, оновіть, подивіться на результат.
2. Опишіть клас ".menu\_item a", який буде описувати стиль посилань, що розміщуються всередині тега, що має клас menu\_item. В класі повинні бути налаштовані наступні параметри: шрифт - Verdana не жирний, текст - не підкреслений, колір посилань #B7F7FF, розмір шрифта 13 пікселів.
3. Створіть клас, який буде при наведенні на посилання змінювати його стан на підкреслене.

***Четвертий рядок головної таблиці призначений для лівого меню і контентної частини документу***

1. Створіть для комірки четвертого рядка клас content\_td, який буде вміщувати тло, що повторюється по осі Х (з файлу bottom.jpg).
2. В цу комірку розташуйте таблиці і призначте коміркам класи:

| class="content\_left"   | class="menu\_table\_top" | | --- | | class="menu\_left" | | class="menu\_table\_bottom" | | class="content\_right" |
| --- | --- | --- | --- | --- |

1. Опишіть кожен клас. Для класа content\_left

.content\_left{

background-image:url(images/palmOS.jpg);

background-position:left bottom;

background-repeat:no-repeat;

padding:8px 8px 290px;

width: 220px;

vertical-align: top;

}

1. Помістіть всередину комірку з класом menu\_left наступний фрагмент сторінки

Уровень первый<br />

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

Уровень второй<br/>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

Уровень третий<br />

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

<a href="#">Ссылка</a>

1. Для класів menu\_table\_top, menu\_left, menu\_table\_bottom опишіть стили так, щоб вони відповідали зображенню на рис.9.1.

menu\_table\_top: высота, ширина, тло;

menu\_left: тло, відступи від границі до тексту, колір тексту, подкреслення тексту, жирність тексту, назва шрифту, розмір шрифту;

menu\_table\_bottom: высота, ширина, тло.

1. Створіть стиль для посилань всередині комірки з класом menu\_left. В назві стиля необхідно вказати назву класа і назву тега. Для цього стиля опишіть властивості згідно з зображенням рис.9.1.
2. Розташуйте в праву контентну частину фрагмент кода з текстового файла (з архіву). Створіть для неї стилі згідно з зображенням рис.9.1.

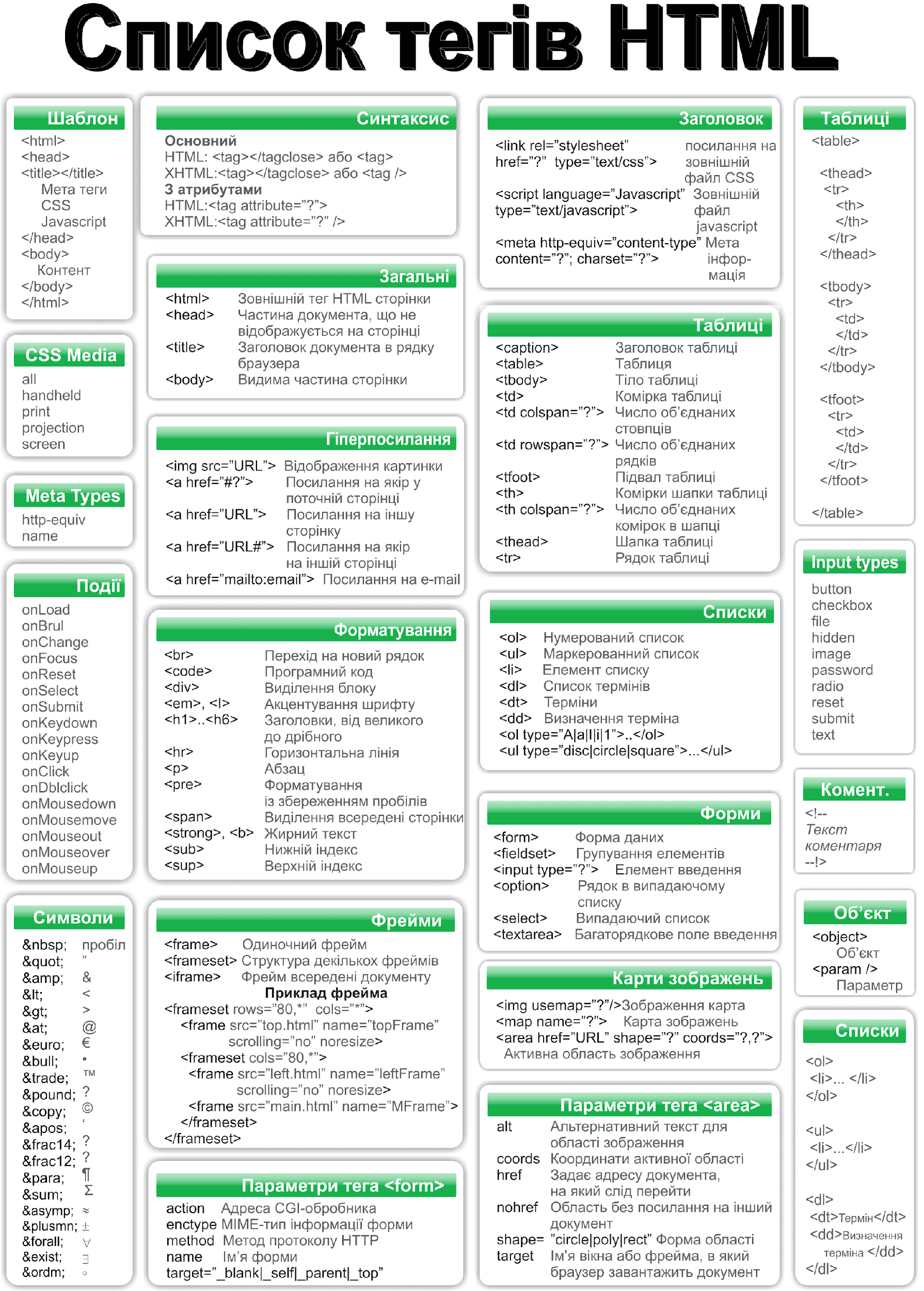
**СПИСОК ЛІТЕРАТУРИ**

1. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В.А.Дронов - СПб.: БХВ-Петербург, 2011. - 416 с.: ил. - (Профессиональное программирование).
2. Росс В. С. Создание сайтов: HTML, CSS, PHP, MySQL/ В.С.Росс [Учебное пособие, ч. 1]- МГДД(Ю)Т, М.:2010 - 107 с.
3. Ташков П. А. Т25 Веб-мастеринг на 100 %: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка /П.А.Ташков. - СПб.: Питер, 2010. - 512 с.: ил. - (Серия «На 100%»).
4. Пасічник О.Г. Основи веб-дизайну /О.Г.Пасічник, О.В.Пасічник, І.В.Стеценко: [Навч.посіб.].-К.:Вид. група BHV.- 2009 .- 336 с.: іл.
5. Гончаров А. Самоучитель HTML/ А.Гончаров – СПб.: Питер, 2002. – 240 с.
6. Зельдман Д. Web-дизайн по стандартам Пер. с англ. Г. П. Ковалева/ Д.Зельдман – М.: НТ Пресс, 2005. – 440 с.
7. Матросов А. В. HTML 4.0/А. В.Матросов, А. О.Сергеев, М. П.Чаунин – СПб.: БХВ-Петербург, 2003. – 672 с.: ил.
8. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство,   
   6-е издание. – Пер. с англ./ Ч.Муссиано, Б.Кеннеди – СПб: Символ-Плюс, 2008. – 752 с., ил.
9. Полонская Е.Л. Язык HTML. Самоучитель/ Е.Л.Полонская – М: Издательский дом “Вильямс”, 2003. – 320 с. : ил.
10. Создание Web-страниц и Web-сайтов. Самоучитель: [учеб. пособие] / Под ред. В. Н. Печникова. – М. : Изд-во Триумф, 2006. – 464 с.: ил.

**ДОДАТОК А**

(довідковий)

**Список тегів HTML**



**ДОДАТОК Б**

(довідковий)

**Розміри**

В CSS існують наступні міри довжини:

**em** - еквівалентні обчисленому значенню властивості 'font-size' того елемента, в якому воно використовується. Виключенням є той випадок, коли 'em' з'являється в самому значенні властивості 'font-size', тоді воно відноситься до розміру шрифту батьківського елемента. Може бути використано для вертикального або горизонтального виміру.

**ex** - визначаються 'x-height' шрифту. x-height називається так, тому що вона часто дорівнює висоті "x" нижнього регістра. Однак 'ex' визначені навіть для тих шрифтів, які не містять "x".

**px** - пікселі відносні до дозволу пристрою перегляду, тобто найчастіше - дисплея комп'ютера. Якщо щільність пікселів пристрою виведення сильно відрізняється від щільності типового комп'ютерного дисплея, ПА повинен перемасштабувати піксельні значення. Рекомендується, щоб піксель як точка відліку був візуальним кутом одного пікселя на пристрої із щільністю пікселів 90dpi на відстані витягнутої руки від читача. При нормальній довжині руки 28 дюймів візуальний кут буде приблизно 0.0227 градусів.

**in** - дюйми - 1 дюйм дорівнює 2.54 сантиметри.

**cm** - сантиметри

**mm** - міліметри

**pt** - пункт, використовуваний в CSS2, дорівнює 1/72 дюйма.

**pc** - піки. 1 піка дорівнює 12 пунктам.

   div { margin: 0.5em } /\* em \*/

   span { margin: 1ex } /\* ex \*/

   P { font-size: 12px } /\* px \*/

   H1 { margin: 0.5in } /\* дюйми \*/

   H2 { line-height: 3cm } /\* сантиметри \*/

   H3 { word-spacing: 4mm } /\* міліметри \*/

   H4 { font-size: 12pt } /\* пункти \*/

   H4 { font-size: 1pc } /\* піки \*/

**ДОДАТОК В**

(довідковий)

**Кольори**

Кольори це або визначене слово, або числова специфікація RGB.

Список назв кольорів: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white й yellow. Ці 16 кольорів визначені в HTML 4.0. Користувач може додатково специфікувати ключові слова, що відповідають кольорам певних об'єктів в середовищі користувача.

   p {color: red}

Формат значення RGB у 16-річному запису - '#' за яким відразу йдуть три або шість 16-річних символів. Трисимвольний запис RGB (#rgb) конвертується в шестисимвольну форму (#rrggbb) шляхом дублювання цифр, але не доповненням нулями. Наприклад, #fb0 розширюється до #ffbb00. Це гарантує, що білий (#ffffff) можна специфікувати скороченим записом (#fff) і видалити залежність від глибини кольорів на дисплеї.

   EM { color: #ff0000 }

Формат значення RGB у функціональному записі - 'rgb(' за яким іде список розділених комами трьох числових значень (або трьох цілих, або трьох процентних) з наступної ')'. Ціле значення 255 відповідає 100% й F або FF в 16-річному запису: rgb(255,255,255) = rgb(100%,100%,100%) = #FFF.

   EM { color: rgb(255,0,0) }

   EM { color: rgb(100%, 0%, 0%) }

**ДОДАТОК Г**

**Варіанти завдань для самостійної роботи**

| Варіант 01. Весільний салон |  | Варіант 02. Професійна рекламна студія |
| --- | --- | --- |
| Варіант 03. Ресторан |  | Варіант 04. Будівельна компанія |
| Варіант 05. Боулінг клуб |  | Варіант 06. Зоопарк |
| Варіант 07. Ювелірний магазин |  | Варіант 08. Магазин антикваріату |
| Варіант 09. Футбольний клуб |  | Варіант 10. Інтернет розробники |
| Варіант 11. Архітектурна компанія |  | Варіант 12. Гостиниця |
| Варіант 13. Забудова котеджів |  | Варіант 14. Бізнес компанія |
| Варіант 15. Кінокомпанія |  | Варіант 16. Магазин одягу |
| Варіант 17. Портфоліо фотографа |  | Варіант 18. Дизайн студія |