

## 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 XAMPP та HTMLPad. XAMPP – це безкоштовне програмне забезпечення з відкритим вихідним кодом, 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-серверу, 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 вебсторінок 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 1 Гб і жорсткий диск, з вільним об’ємом не менше 10 Гб. Операційна система – Windows 2007 і більш пізніші версії, включаючи повну сумісність програмного забезпечення з Windows 11.

# 1 ЛАБОРАТОРНА РОБОТА №1

**Тема:** Web-сервер ХАМРР. Введення в HTML

**Мета:** 1) Навчитися встановлювати Denwer, створювати власні домени і піддомени в програмі ХАМРР.  
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

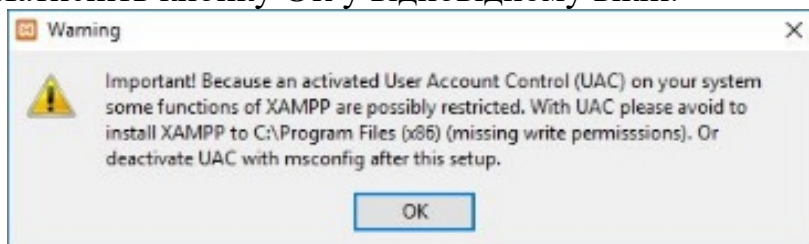
### 1.1.1 ХАМРР. Встановлення програмного продукту.

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

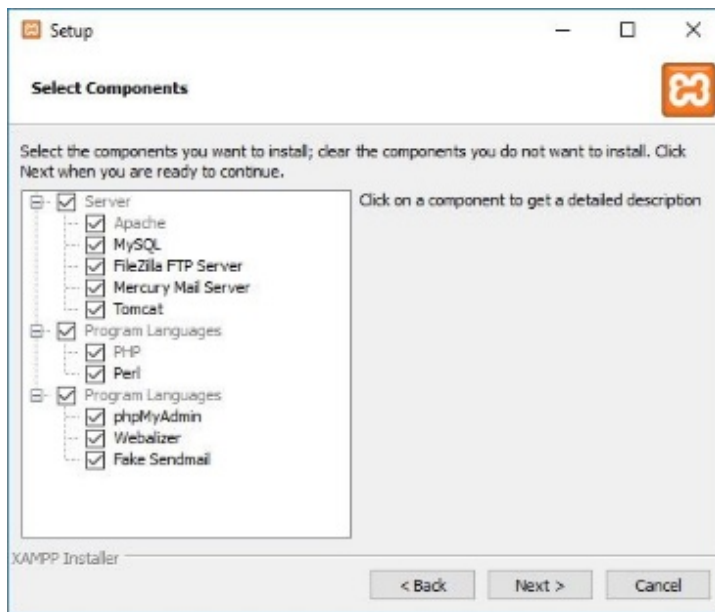
ХАМРР надає веб-дизайнерам і розробникам простий спосіб встановити необхідні компоненти для запуску програмного забезпечення на основі PHP, наприклад WordPress , Drupal, Joomla та інші на Windows 10, macOS та Linux.

Для встановлення програмного продукту ХАМРР необхідно викачати дистрибутив з сайту <https://www.apachefriends.org/download.html> відповідно до вашої операційної системи та слідкуйте наступним пунктам:

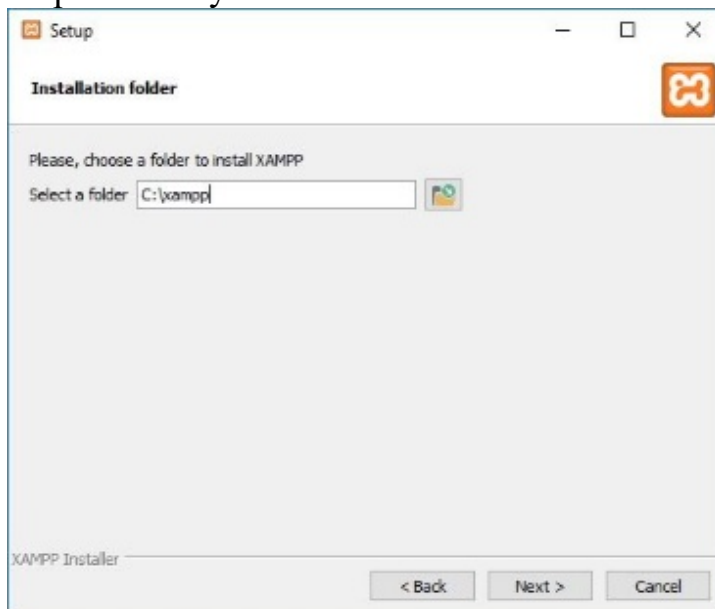
1. Двічі клацніть завантажений файл, щоб запустити програму встановлення.
2. Натисніть кнопку **Ок** у відповідному вікні.



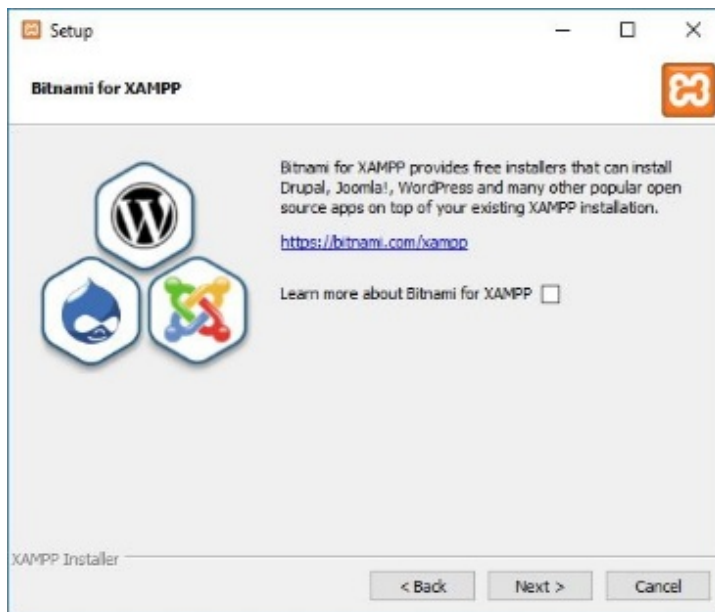
3. ХАМРР пропонує різноманітні компоненти, які ви можете встановити, наприклад MySQL, phpMyAdmin, PHP, Apache тощо. Здебільшого ви будете використовувати більшість цих компонентів, тому рекомендується залишити параметри за замовчуванням.



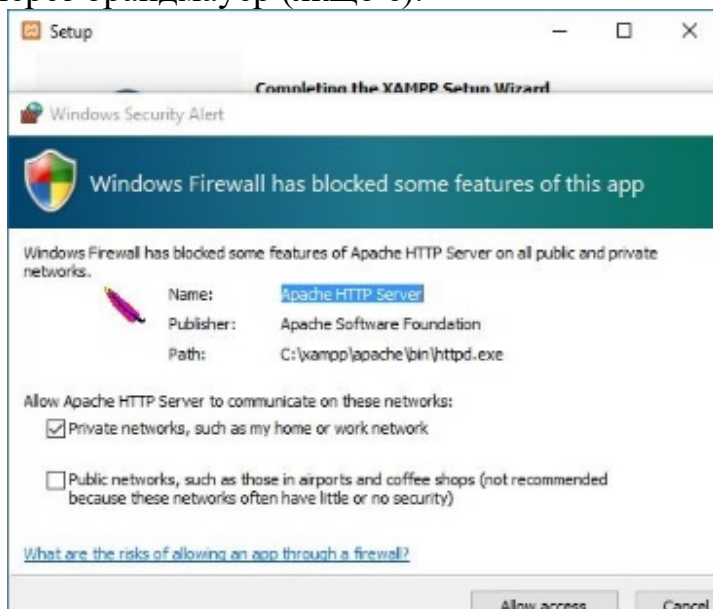
4. Використовуйте місце встановлення за замовчуванням або виберіть іншу папку для встановлення програмного забезпечення в полі Вибрати папку.



5. Приберіть галочку Дізнайтеся більше про Вітнамі для ХАМРР.



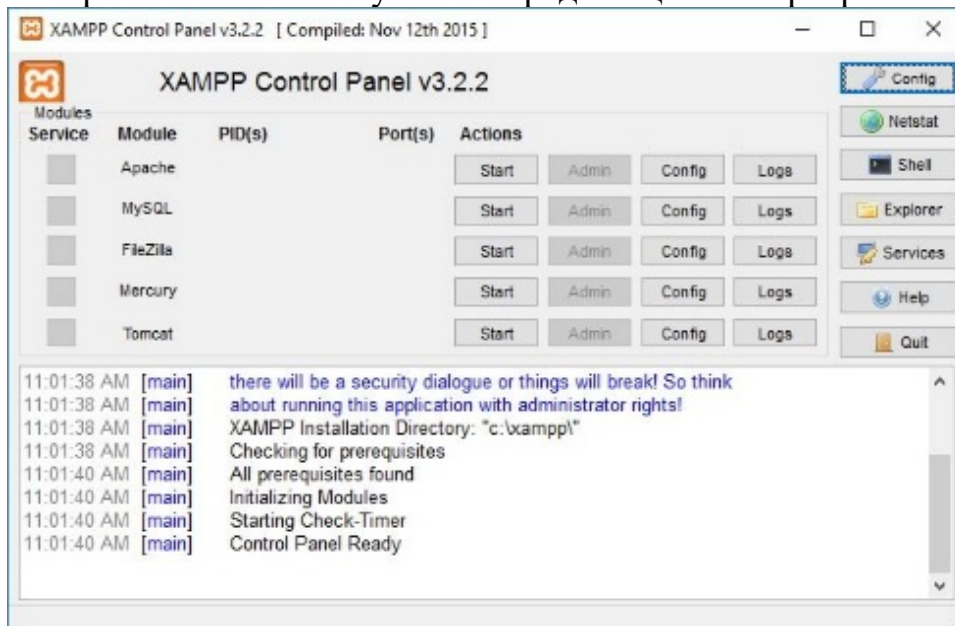
6. Натисніть кнопку **Дозволити доступ**, щоб дозволити програмі пройти через брандмауер (якщо є).



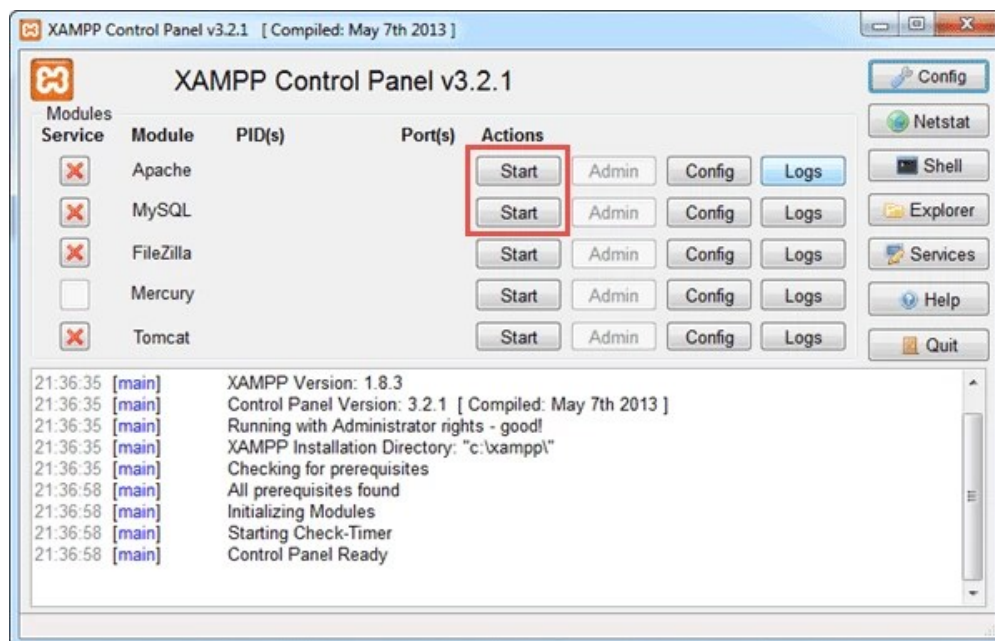
7. Натисніть кнопку **Закінчити**.



Після виконання цих кроків запусниться панель керування XAMPP, і ви зможете розпочати налаштування середовища веб-сервера.



Для початку роботи з сервером досить натиснути кнопки «**Start**» Навпроти сервера Apache і бази даних MySQL. Запущені служби підсвічені зеленим фоном і стають активними їх кнопки «Admin».



На цьому етапі зазвичай надходить повідомлення про систему безпеки від Windows, але ви можете ні про що не турбуватися. Переконайтеся, що ви натиснули на «Розблокувати», і ваш локальний сервер XAMPP буде готовий до використання.

Щоб перевірити, чи все зроблено правильно, відкрийте браузер і перейдіть за цією адресою:

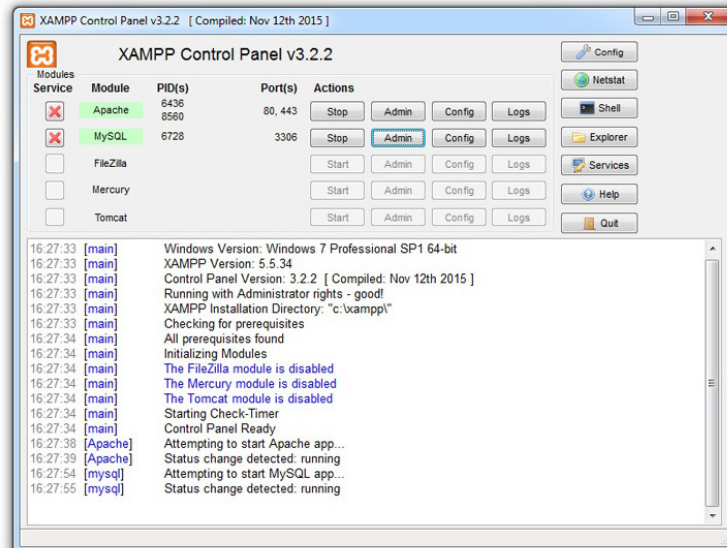
<http://localhost>

Якщо XAMPP налаштований правильно, то ви повинні побачити де що наступне:



[English](#) / [Deutsch](#) / [Francais](#) / [Nederlands](#) / [Polski](#) / [Italiano](#) / [Norwegian](#) / [Español](#) / [中文](#) / [Português \(Brasil\)](#) / [日本語](#)

Тепер у вас є свій **локальний сервер на базі XAMPP!** Далі - справа техніки і знань. Тобто далі потрібно розбиратися як робити сайти. А XAMPP вам допоможе в його тестуванні і відображенні.



Кнопка «Admin» апаха відкриває посилання **http: // localhost / dashboard** / в браузері; тут ви знайдете набір корисних даних: питання і відповіді по XAMPP, документацію і запуск phpinfo (інформація про поточні настройки і конфігурації PHP). Все це поки тільки англійською мовою.

Кнопка «Admin» MySQL запускає phpMyAdmin - додаток для роботи з базою даних.

### 1.1.2 Створення власного домену на локальному комп'ютері

Після встановлення програми усі локальні домени розташовуються в папці C:\xampp\htdocs. 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:\xampp\htdocs. При створенні власного домену слід враховувати неприпустимість використання в імені домену пробілів, букв кирилиці і спеціальних символів.

Якщо у вас є файли сайту для тесту, запустити його на локальному сервері XAMPP досить просто: копіюєте всі файли сайту в окремий каталог в папку сервера, обрану при установці (за замовчуванням це була папка **C:\xampp\htdocs\**).

Наприклад, у вас є папка **sait** з файлами сайту. Після копіювання в локальний сервер шлях до файлів сайту буде наступний: C:\xampp\htdocs\sait\ . А доступ до сайту в браузері буде **http://localhost/sait/**. Тобто, набравши в браузері даний шлях, ви «запросите» у локального сервера файли в папці **sait**, і вам відкриється індексний файл сайту (напр. **index.html**, або **index.php** і т.д.).

Ви можете змінити назву папки свого сайту на будь-який домен, наприклад, **google.com**. Тоді шлях до файлів сайту буде не C:\xampp\htdocs\sait\, а C:\xampp\htdocs\google.com\ . І в браузері потрібно набирати **http://localhost/google.com/**.



Наявність в адресному рядку шляху **localhost** трохи заплутує, але служить для нагадування, що ви працюєте на локальному сервері, а не на віддаленому хостингу. І додасть вам трохи нервів при перенесенні сайту на робочий хостинг (потрібно буде замінити в файлах і / або базі даних всі статичні посилання з **localhost** в адресі), але це вже тема для окремого уроку.

Після завершення робочого дня і тестування сайту, не забувайте зупиняти роботу локального сервера XAMPP шляхом натискання на кнопки «Stop» в панелі адміністрування.

Деякі моменти:

- Локальний сервер XAMPP підійде вам, якщо у вас не встановлено Windows, а наприклад, Linux або Mac OS. Денвер працює тільки на операційній системі Windows.
- Це досить спірний момент, але завантаження і обробка сайтів на складних CMS (наприклад, маджента), в XAMPP відбувається трохи швидше, ніж в Денвері. На цю швидкість, все ж, більше впливає конфігурація комп'ютера - скрипти і код обробляються швидше, якщо у вас досить потужна машина.
- Відсутність російської локалізації, документації російською, робить XAMPP не таким популярним, як Денвер в Східній Європі.
- Можливі конфлікти при одночасній роботі Skype і XAMPP через використання в Skype 80 порту.
- XAMPP і Денвер є безкоштовними, з відкритим вихідним кодом, але для скачування дистрибутива Денвер знадобиться реєстрація.

### 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 2020». 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.



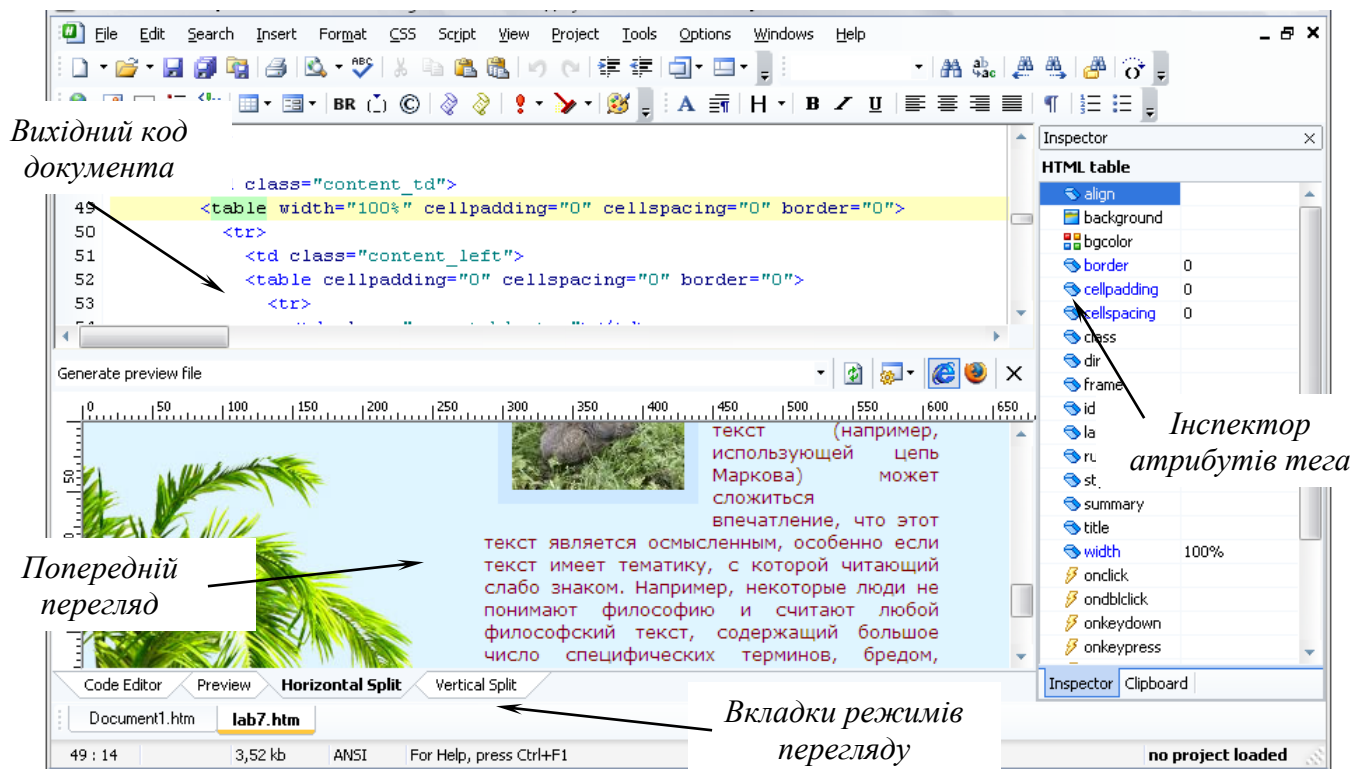


Рис.1.3. Інтерфейс програми “HTML Pad 2020”

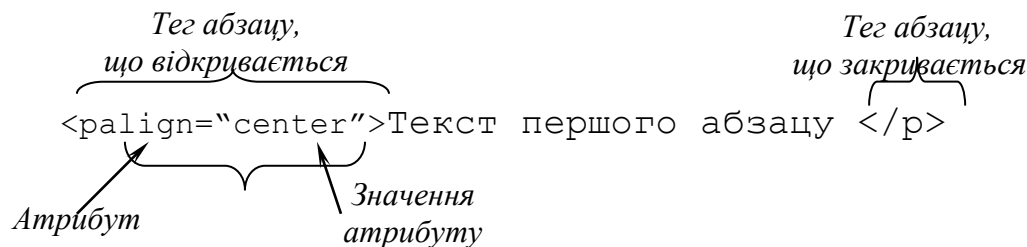
Після створення нової сторінки основні теги можна додавати вручну, або через пункти меню Insert (Вставити) і Format (Форматування), або за допомогою кнопок на панелях інструментів HTML, HTML Formatting та HTML Forms.

Перемикається між режимом редагування і режимом попереднього перегляду можна також за допомогою використання вкладок перемикавання 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**.

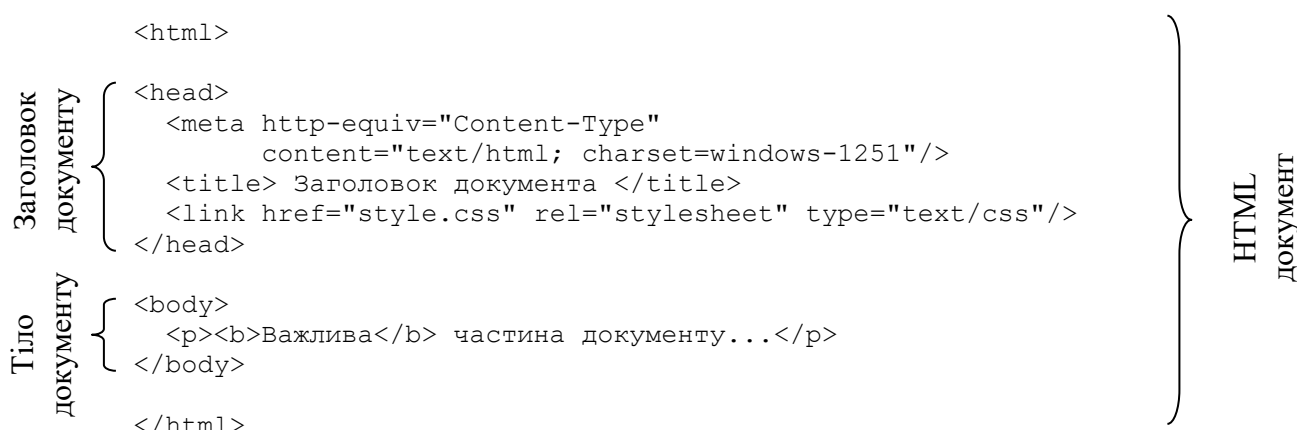
Основною структурною одиницею мови є тег з його атрибутами. Тег являє собою розміщені в трикутних дужках інструкції мови, які повідомляють, якими властивостями повинен володіти той або інший фрагмент тексту на сторінці. Більшість тегів є парними: містять як відкриваючий, так і закриваючий тег (див. рис.1.4):



**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).



**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 вебсторінку. 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 вебсторінки (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 в цілому.

Основні атрибути тега &lt;body&gt;

Назва атрибуту	Призначення атрибуту
alink	колір активного посилання (колір задається шістнадцятиричним числом або константою, наприклад червоний: red або #ff0000)
background	фоновий рисунок на вебсторінці
bgcolor	колір фону вебсторінки
topmargin(leftmargin, bottommargin)	відступ від верхнього (лівого, нижнього) краю вікна браузера до контенту
link	колір посилань на вебсторінці
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 &lt;body&gt;

```

1  <html>
2
3  <head>
4    <title>Лістинг 1.1</title>
5  </head>
6
7  <body bgcolor="#FFFF33" text="#009900" link="black">
8    Текст документа. <a href="#">Посилання</a>
9  </body>
10
11 </html>

```

### 1.1.6 Основні теги форматування тексту

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

1	2
<code>&lt;code&gt;...&lt;/code&gt;</code>	призначений для відображення тексту, який є програмним кодом, зазвичай відображується моноширинним шрифтом
<code>&lt;del&gt;...&lt;/del&gt;</code>	використовується для виділення тексту, який був видалений в новій версії документа
<code>&lt;ins&gt;...&lt;/ins&gt;</code>	використовується для виділення тексту, який був доданий в нову версію документа
<code>&lt;dfn&gt;...&lt;/dfn&gt;</code>	застосовується для виділення термінів при їх першій появі в тексті
<code>&lt;sup&gt;...&lt;/sup&gt;</code>	відображує шрифт у вигляді верхнього індексу
<code>&lt;sub&gt;...&lt;/sub&gt;</code>	відображує шрифт у вигляді нижнього індексу
<code>&lt;pre&gt;...&lt;/pre&gt;</code>	визначає блок тексту, в якому зберігається кількість пропусків між словами, задана на етапі створення документа
<code>&lt;nobr&gt;...&lt;/nobr&gt;</code>	повідомляє браузер відображувати текст в один рядок без перенесення
<code>&lt;h1&gt;...&lt;/h1&gt;</code> ... <code>&lt;h6&gt;...&lt;/h6&gt;</code>	встановлює заголовки різного рівня значущості від найкрупнішого h1 до найдрібнішого h6
<code>&lt;font&gt;...&lt;/font&gt;</code>	призначений для встановлення характеристик шрифту, які задаються в атрибуті тега. Тег має наступні атрибути: <b>color=...</b> задає колір шрифту (у шістнадцятиричній формі, наприклад, #ff0000) <b>face=...</b> задає гарнітуру шрифту (Arial, Tahoma) <b>size=...</b> задає розмір шрифту в умовних одиницях (ціле число від 1 до 7 або зміни числа +1 або -2) найкрупніший шрифт 7
<code>&lt;p&gt;...&lt;/p&gt;</code>	визначає межі абзацу. Тег має атрибут <b>align</b> , який відповідає за вирівнювання тексту і може набувати значень: <b>left</b> , <b>right</b> , <b>center</b> , <b>justify</b>
<code>&lt;br&gt;</code>	встановлює примусове перенесення рядків
<code>&lt;hr&gt;</code>	малює горизонтальну лінію. Параметри горизонтальної лінії можна задати з використанням атрибутів тега: <b>align=...</b> визначає вирівнювання лінії <b>color=...</b> колір лінії <b>noshade=...</b> малює не тривимірну лінію <b>size=...</b> товщина лінії в пікселях <b>width=...</b> 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. натиснути 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 XAMPP.
2. Для створення власних доменів, на локальному комп'ютері створіть в папці C:\xampp\htdocs\ папку з ім'ям домену **help.ua**, на якому ви будете проводити тестування сайту. Сам сайт знаходиться в архіві, що додається до лабораторної роботи.
3. У папку **help.ua** в яку скопіюйте вміст папки **htmlbook** (узятую з архіву до лабораторної роботи).
4. Перевірте працездатність сайту через браузер за адресою <https://localhost/help.ua>.
5. Створіть свій домен, в імені якого буде присутнє ваше прізвище і група. У папці вашого каталогу створіть файл **index.html**. Саме з цього файлу завжди починається завантаження вмісту домену, якщо не вказаний інший файл.
6. Відкрийте файл **index.html** за допомогою програми HTML Pad 2020.
7. Запишіть в ньому наступні рядки:

```
<html>
  <head>
    <title> Заголовок вікна </title>
  </head>
  <body>
    Освоюємо web-технології :)
  </body>
</html>
```

← Заголовок вікна браузера

} Те, що буде відображатися всередині вікна браузера

8. Збережіть файл. Перевірте в браузері результат, набравши в адресному рядку ім'я вашого сервера із завдання 5. У разі виникнення проблем із завантаженням сервера зверніться до теоретичних відомостей.
9. 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).
10. 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.

11. Всі файли і зображення, які з'являтимуться на вашій вебсторінці, повинні розташовуватися всередині папки **<ім'я домена>/** і мати адреси починаючи з кореня сайту, а не з кореня диска C: або D: Виберіть фонове зображення, скопіюйте його в папку **<ім'я домена>** і змініть фон сторінки (атрибут **background** в тезі **body**). Збережіть і прогляньте результати роботи в браузері.
12. 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 13). При форматуванні використовуйте теги з теоретичних відомостей, а також детальний опис тегів з домена **html.ua**;
13. Створіть новий документ з ім'ям **lab1.html** і збережіть його в корені свого домену.
14. Між відкриваючим та закриваючим тегом **<body>** додайте текст з файлу. Збережіть його і прогляньте в браузері що вийшло.
15. Змініть фон сторінки і колір тексту згідно рисунку.



Найкращий спосіб у чомусь розібратися до кінця – це спробувати навчити цьому комп'ютер.

USB (англ. Universal Serial Bus<sup>1</sup>) – універсальна послідовна шина, призначена для підключення периферійних пристроїв. Шина USB є послідовний інтерфейс для передачі даних для середньошвидкісних та низькошвидкісних периферійних пристроїв.

Для підключення периферійних пристроїв до шини USB використовується чотирипровідний кабель, при цьому два проводи (віта пара) у диференціальному включенні використовуються для прийому та передачі даних, а два дроти - для живлення периферійного пристрою. Завдяки вбудованим лініям живлення, USB дозволяє підключати периферійні пристрої без власного джерела живлення (максимальна сила струму, який споживає пристрій по лініях живлення шини USB не повинна перевищувати 500 мА). Попередні версії

USB 0.7: специфікація випущена листопаді 1994 року

USB 0.8: специфікація випущена у грудні 1994 року

USB 0.9: специфікація випущена у квітні 1995 року

USB 0.99: специфікація випущена у серпні 1995 року

USB 1.0 Release Candidate: специфікація випущена у листопаді 1995 року

USB 1.1 специфікацію випущено у вересні 1998 року. Виправлені проблеми та помилки, виявлені в Версії 1.0. Перша версія, що набула масового поширення

USB 2.0 специфікація випущена у квітні 2000 року

## Пісня про DOS

DOS. Чорний пеленою екран заповнив чистий DOS  
Миша. Втратила форму, раптом стала квадратна миша  
Я розбив вікно, дев'яносто п'яте вікно мастда  
І поставив DOS, і тоді я зрозумів: Це щастя – ось воно

Краплі на окулярах. Дивні окуляри, а може, сльози на обличчі.  
DOS очистив все, що було зайвим у мене на диску C.  
Я натиснув F8 та веселий Norton видаляв мені все поспіль  
40 мегабайт, може навіть більше, може навіть 60...

І уявив я: місто наповнилося раптом розумними людьми  
Вийшли всі під DOS, а проклятий Windows видаляли, чорт забирай  
Забувши про Word, MS Excel, Corel Draw та інший геморої  
Люди ставлять DOS. Немов у Рай заходять до DOS, нормальний, чистий DOS

Приклади пошукових систем: [Google.com](http://Google.com), [ya.ru](http://ya.ru).

## Які результати виведе шматок програмного коду на екран?

```
int x = 1, z = ++x + x;  
Console.WriteLine (x.ToString () + " " + z.ToString ());  
  
x = 1;  
int z1 = x + ++x;  
Console.WriteLine (x.ToString () + " " + z1.ToString ());  
  
Лабораторну роботу виконав © Студент acontadini™
```

Рис.1.6. Зразок для виконання практичного завдання

16. Розбийте текст на абзаци використовуючи тег **<p>**.
17. Зверніть увагу на те, щоб у вас не було червоних (не закритих) тегів.
18. 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"**.

19. Встановіть словам жирне, підкреслене і курсивне накреслення.
20. У другому абзаці використайте верхній індекс.
21. Використовуючи спеціалізовані теги виділіть цитату і програмний код. Використовуйте додатковий тег **<pre>** для збереження числа пропусків між словами в програмному коді.
22. Використовуючи тег **<font>** виділіть назви пошукових систем.
23. Змініть колір яндекса на жовтий.
24. Додайте в документ дві лінії, першу завдовжки в 400 пікселів, а заввишки в 5 пікселів. А другу і третю шириною 50%.
25. Використовуючи теги **<h1>...<h6>** виділіть заголовки.
26. Використовуючи коди спеціальних символів відобразіть на сторінці знак менше. І підпишіть виконане завдання як вказано на рисунку 1.6.
27. 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.
28. Перевірте, що всі файли у вас доступні через сервер з використання доменного імені.
29. Покажіть виконану роботу викладачу.

### 1.3 Контрольні питання

1. Як додати до XAMPP 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 2020?
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 Inspectrot вказати його атрибути. Можна також скористатися кнопкою Image на панелі інструментів HTML програми HTMLPad 2020.

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 i 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

## 2.1.2 Списки

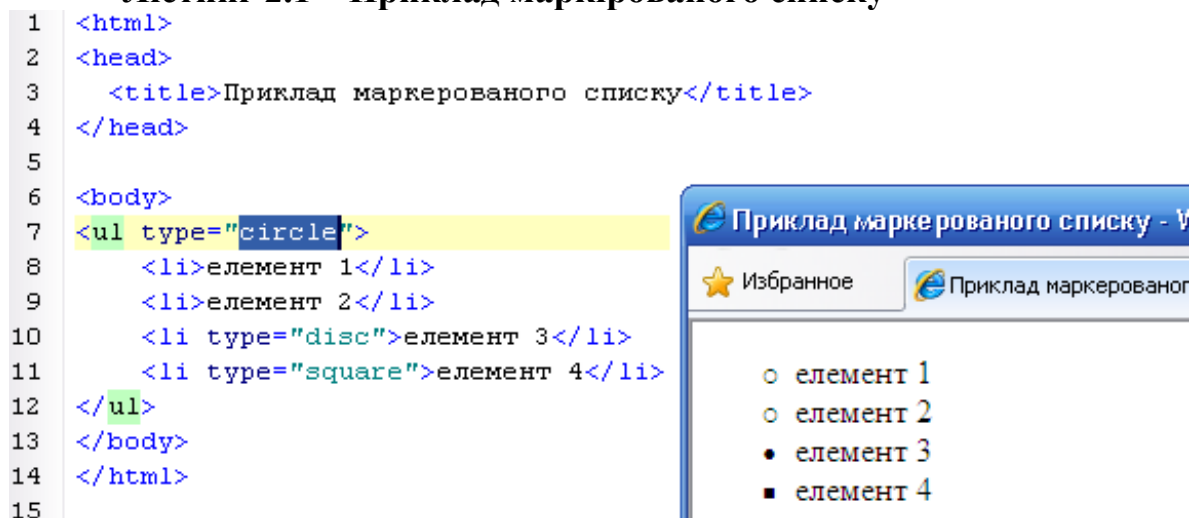
У мові HTML розрізняють наступні види списків:

- 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 елемента списку.

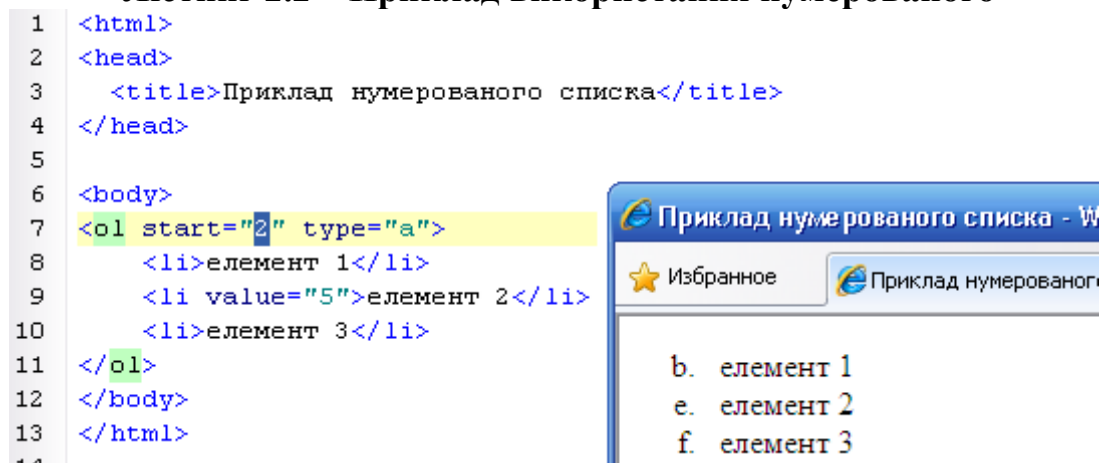
За замовчуванням елементи списку маркуються чорним кружечком. За допомогою атрибуту **type** можна змінити стиль маркування. Приклад використання списку наведений в Лістингу 2.1:

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:

### **Елемент 1**

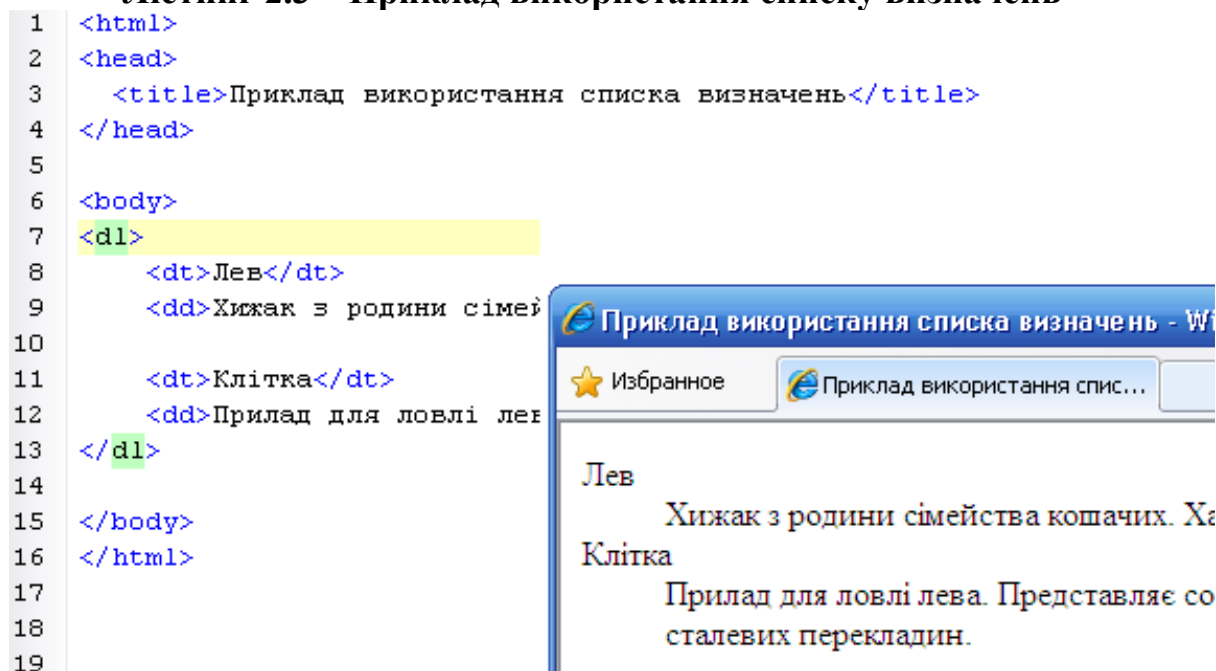
Визначення елементу 1

### **Елемент 2**

Визначення елементу 2

Сам список задається за допомогою тега **<dl>**, елемент – 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.

## Атрибути тега &lt;a&gt;

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

```

1 <html>
2 <head>
3   <title>Тег А, атрибут name</title>
4 </head>
5
6 <body>
7   <p><a name="top"></a>
8   В эту точку будет осуществлен переход при щелчке на нижнюю ссылку</p>
9
10  <a href="http://google.com" title="" target="_blank"
11     title="ссылка откроется в новом окне">Поисковая система google</a>
12
13  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit,
14     tincidunt ut lacreet dolore magna aliquam erat volutpat.</p>
15
16  <p><a href="#top">Наверх</a></p>
17 </body>
18 </html>

```

## 2.2 Практична частина

1. Запустіть веб-сервер ХАМРР.
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>. Змініте 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 сьогоднішньої лабораторною як іменованій якір: **<a name="image">**, **<a name="spisok">**.
9. На початку сторінки додайте список посилань на всі розділи лабораторної роботи. Пам'ятаєте, що при посиланні на іменованій якір слід використовувати знак # перед ім'ям якоря.
10. Додайте внизу документа посилання на свою сторінку у FaceBook, яка відкриватиметься в новому вікні.
11. Друге зображення в документі зробіть посиланням на пошукову систему Google
12. Створіть новий файл **index.html**, у якому розробіть список, що містить посилання на завдання двох лабораторних робіт.
13. Створіть новий файл **my.html** і запишіть в ньому: нумерований список з відомих вам мов програмування, маркірований список з відомих вам палітр кольорів.
14. Додайте посилання на файл **my.html** у файл **index.html**.
15. 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 межею довгий час використовувалися для верстки вебсторінок, 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
align	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
cellpadding	відступ від рамки до вмісту клітинки

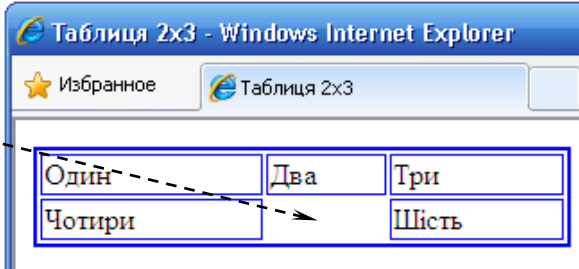
При побудові таблиця розділяється на рядки (тег **<tr>**), які у свою чергу діляться на клітинки (тег **<td>**). Приклад таблиці наведений в лістингу 3.1.

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

```

1  <html>
2  <head>
3    <title>Таблиця 2x3</title>
4  </head>
5  <body>
6    <table width="300" border="2" bordercolor="#0000FF">
7      <tr>
перший рядок { <td>Один</td>
                <td>Два</td>
                <td>Три</td>
11             </tr>
12             <tr>
другий рядок { <td>Чотири</td>
                <td></td>
                <td>Шість</td>
16             </tr>
17           </table>
18 </body>
19 </html>

```



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

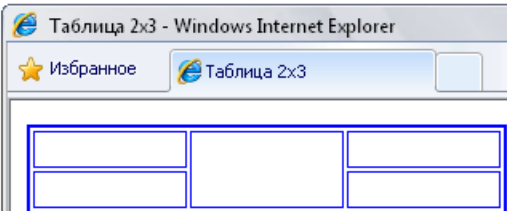
Для об'єднання елементів таблиці використовують атрибути **colspan** – для вказівки числа об'єднаних клітинок по горизонталі і **rowspan** – для вказівки числа об'єднаних клітинок по вертикалі. Приклад об'єднання елементів таблиці приведені в лістингах 3.2 і 3.3.

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

```

6  <table width="300" border="2" bordercolor="#0000FF">
7    <tr>
8      <td>&nbsp;</td>
9      <td rowspan="2">&nbsp;</td>
10     <td>&nbsp;</td>
11   </tr>
12   <tr>
13     <td>&nbsp;</td>
14     <td>&nbsp;</td>
15   </tr>
16 </table>

```

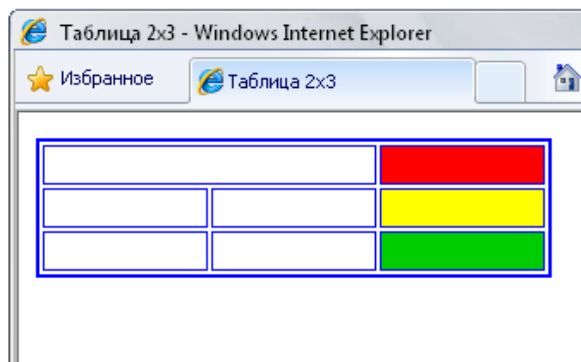


### Листинг 3.3 – Об'єднання клітинок по горизонталі

```

6 <table width="300" border="2" bordercolor="#0000FF">
7   <tr>
8     <td colspan="2">&nbsp;&nbsp;&nbsp;</td>
9     <td bgcolor="#FF0000">&nbsp;&nbsp;&nbsp;</td>
10  </tr>
11  <tr>
12    <td>&nbsp;&nbsp;&nbsp;</td>
13    <td>&nbsp;&nbsp;&nbsp;</td>
14    <td bgcolor="#FFFF00">&nbsp;&nbsp;&nbsp;</td>
15  </tr>
16  <tr>
17    <td>&nbsp;&nbsp;&nbsp;</td>
18    <td>&nbsp;&nbsp;&nbsp;</td>
19    <td bgcolor="#00CC00">&nbsp;&nbsp;&nbsp;</td>
20  </tr>
21 </table>

```



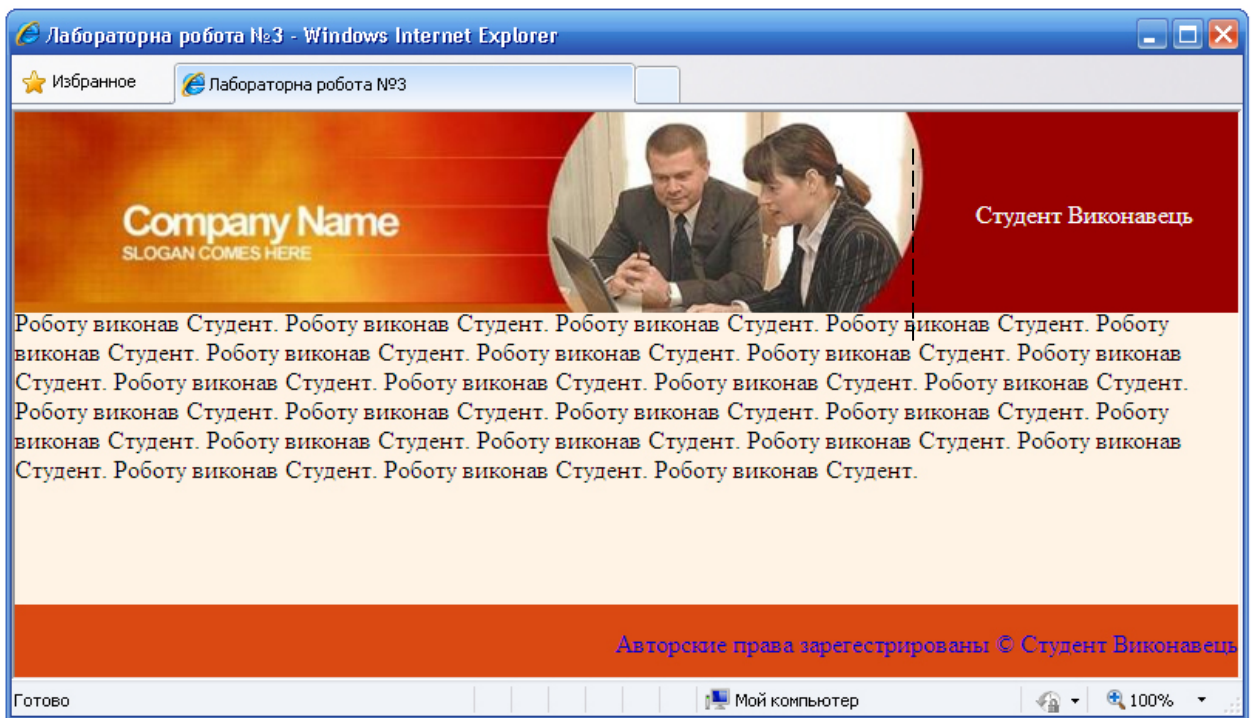
Приклад створення таблиці для створення сайту, буде розглянутий в практичній роботі

## 3.2 Практична частина

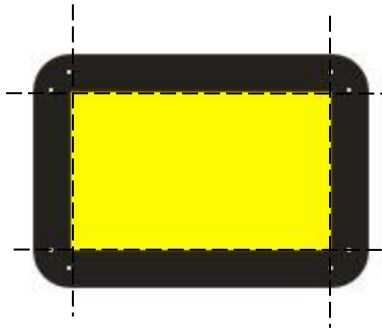
1. Запустіть веб-сервер ХАМРР.
2. Створіть файл **lab3.htm** і збережіть його в своїй папці на веб-сервері.
3. Створіть у файлі **lab3.htm** таблицю розміром 3x3 для гри в «хрестики-нулі». Для створення таблиці використовують тег **<table>**. Для таблиці необхідно створити рамку для відображення меж. 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.

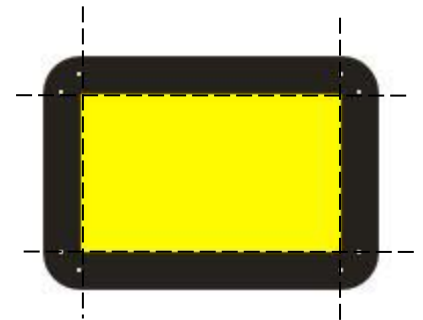
Стать	Середні показники	
	Зріст	Вага
Чоловік	174	78
Жінка	165	65


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 браузера 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 ©.
7. У окремому файлі створити таблицю розміром 3x3 (пунктиром вказані кордони клітинок).
8. 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 відображатиметься 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).

9. Додайте у файл **index.htm** посилання на виконані завдання з лабораторної роботи №3.
10. 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 відображатиметься 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 навички, отримані при створенні таблиць.

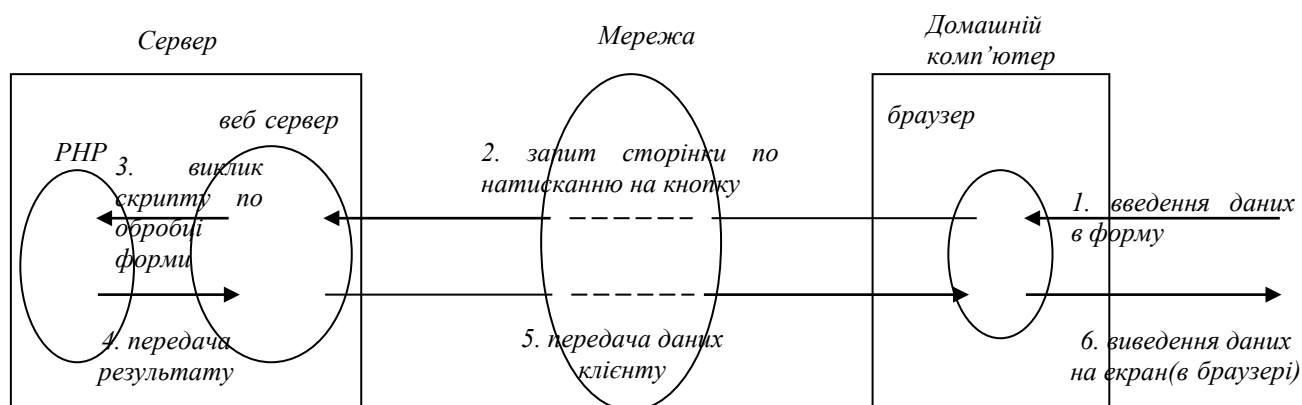
**Час:** 2 год.

### 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: <i>GET</i> – спосіб передачі, коли дані, введені у форму додаються в рядок запиту та <i>POST</i> – дані пердаються у прихованому вигляді

name	ім'я форми. Використовується для звернення до форми за допомогою javascript
target	ім'я вікна або фрейма, куди обробник завантажуватиме повернений результат
enctype	тип інформації форми. Розрізняється в разі використання символічно-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 елемента, – **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 елемента 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 <b>type="image"</b>
alt	визначає альтернативний текст для кнопки із зображенням ( <b>type="image"</b> )
checked	заздалегідь активований перемикач або прапорець
disabled	блокує доступ і зміну елемента (поле недоступне)
maxlength	максимальна кількість символів, дозволених в тексті



name	ім'я поля, призначене для того, щоб відрізнити один елемент керування від іншого. Ім'я може бути вказане по всім правилам іменування змінних
readonly	дозволяє лише читання з текстового елемента керування
size	ширина текстового поля в символах
value	значення елемента. Значення змінної вказаної в атрибуті <b>name</b>

☒ Жіночий  
☐ Чоловічий

### Перемикач

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

### Прапорець ☒ Підписатися на розсилку

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

Підписатися

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

Обзор...

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>**

Варіант два

Оберіть варіант

Варіант один

Варіант два

Варіант три

`</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

```

1 <html>
2 <head>
3   <title>Приклад форми</title>
4 </head>
5
6 <body>
7 <form name="info" action="http://myserver.ru/obrabotka.php"
8   method="post" enctype="multipart/data">
9   Прізвище та по-батькові:
10    <input name="name" type="text" size="20" /><br />
11   Відповідь на секретне запитання:
12    <input name="passwords" type="password" /><br />
13    <input type="checkbox" name="zapros" checked="checked" />
14   Надіслати пароль на пошту<br />
15   Ксерокопія паспорта:
16    <input name="" type="file" /><br />
17   Вкажіть вашу стать: Ч <input type="radio" name="variant" />
18   Ж <input type="radio" name="variant" /><br />
19   У яких соціальних мережах ви зареєстровані?<br />
20    <select name="seti" size="3" multiple="multiple">
21      <option>Odnoklassniki</option>
22      <option>Мой мир</option>
23      <option>Вконтакте</option>
24    </select><br />
25   Додаткова інформація про себе:<br />
26    <textarea name="about" rows="4" cols="30"></textarea><br />
27    <input type="submit" name="send" value="Запросить пароль" />
28 </form>
29 </body>
30 </html>

```

Приклад форми - Windows Internet Explorer

Избранное Приклад форми

Прізвище та по-батькові:

Відповідь на секретне запитання:

☒ Надіслати пароль на пошту

Ксерокопія паспорта:

Вкажіть вашу стать: Ч ☐ Ж ☐

У яких соціальних мережах ви зареєстровані?

- Odnoklassniki
- Мой мир
- Вконтакте

Додаткова інформація про себе:

## 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>**.

The screenshot shows a web browser window titled 'Лабораторна робота №4 - Windows Internet Explorer'. The address bar shows 'C:\Documents and Settings\strashilka\Рабо-'. The form is titled 'Лабораторна робота №4' and contains the following fields and controls:

- Ваше ім'я:
- Пароль:
- Дата реєстрації:
- Стать: Ч ☐ Ж ☐
- ☐ Так, я бажаю отримувати новини з вашої організації
- Скільки листів в день ви звичайно отримуєте? ☐ Один-два ☒ До 10 ☐ Дуже багато
- Вкажіть, яким комп'ютерним напрямком ви зацікавлені: 

Комп'ютерна графіка  
Проектування  
Програмування  
Набір тексту  
Ігри  
Спілкування в мережі
- Ваші побажання до роботи поштової системи:
- Аватарка:
- 

The status bar at the bottom shows 'Готово' and 'Мой компьютер'.

Рис.4.2. Форма заповнення анкети

4. Створити поля для введення імені користувача і пароля користувача (символи вводяться в поле Пароль повинні відображатися зірочками). Вставка полів здійснюється тегом **<input>** з різними значеннями атрибуту **type**.
5. 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>**.

6. Створити перемикачі для вибору статі. Реалізувати можливість вибору лише одного варіанту. Використовувати тег **<input>**.
7. Створити прапорець, з текстом «Так, я бажаю отримувати спам» з прапорцем, встановленим за умовчанням. Використовувати тег **<input>**.
8. Створити перемикачі для вибору кількості листів отримуваних в день, перевірити щоб перемикачі із завдання 6 і 8 формували дві різні групи.
9. Створити список інтересів з можливістю вибору декількох елементів.
10. 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>**.
11. Створити поле з кнопкою «Огляд» для завантаження файлу зображення. Використовувати тег **<input>**.
12. Створити дві кнопки. Першу для передачі даних форми на сервер, а другу для очищення форми. Використовувати тег **<input>**.
13. 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**?

## ЛАБОРАТОРНА РОБОТА №5

- Тема:** Фрейми (iframe) у HTML 5. Карти зображень
- Мета:** 2) Навчитися створювати фрейми (iframe);  
3) Навчитися управляти властивостями фреймів(iframe);  
4) Навчитися створювати і застосовувати карти зображень для непрямокутних посилань.
- Час:** 2 год.

### 5.1 Теоретичні відомості

#### 5.1.1 Фрейми у HTML 5

В цій роботі розглянемо сучасний стандарт фреймів, який підтримується сучасною сертифікацією. Але для початку згадаємо, що таке фрейми:

Фрейм – це окреме вікно у html сторінці, в якому відображається інша сторінка.

Фрейм – це «сторінка у сторінці», тобто це сторінка, яка завантажується у середині іншої сторінки.

Фрейми дозволяють розбивати веб-сторінки на скролюємі 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.

Зазвичай фрейми використовують для наступних цілей:

- вбудовування мапи від Google;
  - вбудовування медіаконтенту (у більшості випадків – відео з Youtube)
  - вбудовування прикладів фрагментів коду;
  - вбудовування аплетів додатків, працюючих у контексті веб-сайту.
- Наприклад форми оплати через інтернет банкінг.

Не рекомендується використовувати фрейми у якості:

- фотогалереї;
- вікна чату;
- навігації;
- накрутки фонового трафіку.

Переваги фреймів:

- швидке завантаження відео, мап, презентацій та іншого контенту;
- фрейми у HTML дозволяють виводити одночасно на екран користувачу декілька сторінок, які повністю самостійні;
- користувач може переключати сторінки у фреймах, без перезавантаження основної сторінки.
- дозволяє реалізувати розбиття структури сторінки на блочну структуру, та в разі потреби редагувати виключно потрібні блоки (скорочує деякий час корегування, якщо одна ж і та ж сама інформація розміщена на багатьох сторінках).

Недоліки фреймів:

- зміст фреймів не індексується на сторінці, тому таким чином пошукова індексація сторінки може значно погіршитися в пошукових системах (особливо коли фрейми займають більшу частину сторінки).
- при розміщенні у фреймі інформації зі стороннього ресурсу, у разі подальшої появи на ньому контенту виключно для повнолітніх, ваша сторінка, або взагалі весь сайт також може потрапити під «санкції» від Google, і Ваша сторінка буде значно гірше показуватись (рідше та на значно нижчих позиціях) по деяким запитам, а по деяким може не відображатися взагалі.

«Google підтримує фрейми та iframe в той мірі, в якій це можливо. iFrame можуть створювати проблеми для пошукових систем, тому що вони не відповідають концептуальній моделі інтернет», саме так один з представників гугл заявив на сайті технічної підтримки Google (support.google.com).

Причина, з якої iFrames не відповідають концептуальній моделі, полягає у тому, що сторінки с iFrame мають більше однієї URL-адреси. Одна – для сторінки, на якій розміщується вставка iFrame, кожна наступна – для сторінки, з якої зволікається контент. Недивно, що роботи можуть заплутатись.

У HTML не підтримуються тег <frameset>, але у ньому присутній схожий за призначенням тег: <iframe>

Тег <iframe> створює плаваючий фрейм, котрий знаходиться всередині звичайного документа, він дозволяє завантажувати у область заданих розмірів будь-які інші незалежні документи.

Тег <iframe> є контейнером, зміст якого ігнорується браузером, які не підтримують даний тег. Для таких переглядачів можна вказати альтернативний текст, котрий побачать користувачі. Він повинен бути розміщеним між елементами <iframe> ... </iframe>

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

**Таблиця 5.1 – Атрибути тега <iframe>**

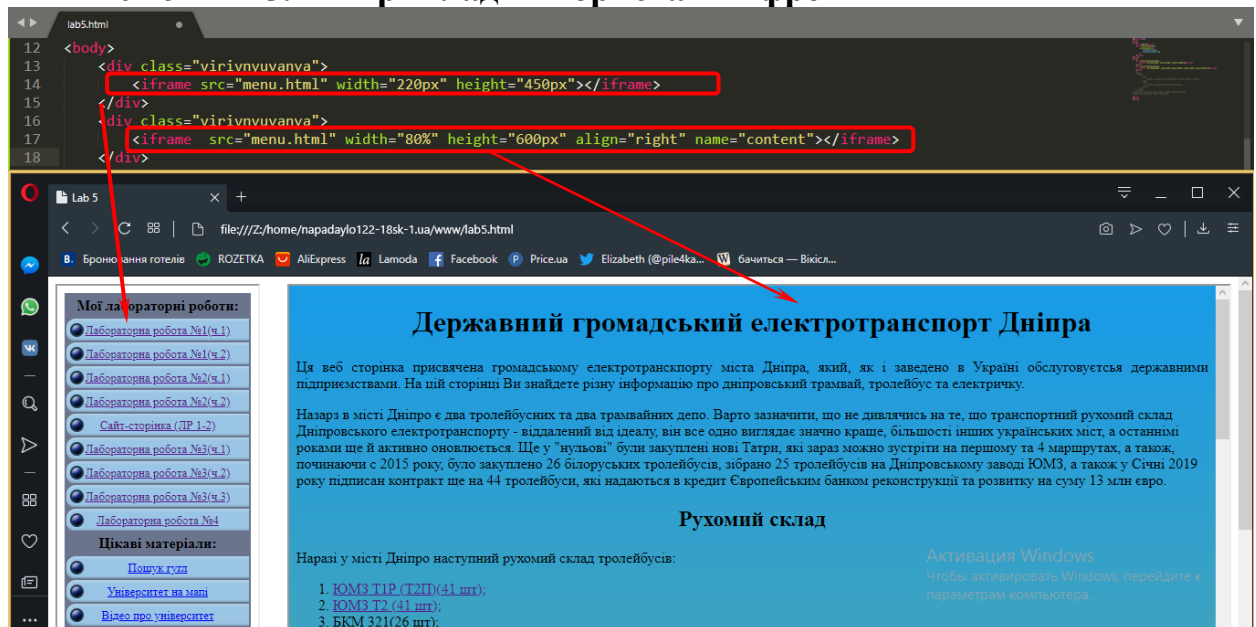
Атрибут	Призначення
src	задає шлях до файлу, призначеного для завантаження у фрейм
name	задає унікальне ім'я фрейма
scrolling	спосіб відображення смуги прокрутки у фреймі. Можливі варіанти значень: <b>yes, no, auto</b>
frameborder	визначає чи слід відображати рамку довкола фрейма
align	Вирівнювання фрейму по горизонталі та/або спосіб обтікання його текстом.
allowtransparency	Встановлює прозорий фон фрейму, через котрий видно фон сторінки.
height	Висота фрейму. За замовчанням: 150px
hspace	Горизонтальний відступ від фрейму до зовнішнього елемента
marginheight	Відступ зверху та знизу від інших елементів до меж фрейму.
marginwidth	Відступ зліва та праворуч від інших елементів до меж фрейму.
sandbox	Дозволяє задати ряд обмежень на контент у фреймі
scrolling	Відображення прокрутки у фреймі. Значення: auto, yes, no.
seamless	Відображення зовнішнього вигляду фрейму, як частини документа.
srcdoc	Зміст фрейму (можливо вбудування фрейму, не тільки з зовнішньої сторінки, але і у середині цього атрибуту у рамках однієї сторінки).



vspace	Вертикальний відступ від фрейму до зовнішнього елемента
width	Ширина фрейму. За замовчанням: 300px

Приклад використання фреймів розглянутий в лістингу 5.1:

### Лістинг 5.1 – Приклад використання фреймів



Коментарій до лістингу.

1. у лівому фреймі відкритий файл menu.htm і посилання цьому файлу організовані з вказівкою цільового фрейма, в якому повинні відкривати нові файли;
2. 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>

```
<a href="lab1 new.html" target="content">Лабораторна робота №1(ч.1)</a></br>
<a href="lab1.1.html" target="content">Лабораторна робота №1(ч.2)</a></br>
<a href="lab2.htm" target="content">Лабораторна робота №2(ч.1)</a></br>
<a href="my.html" target="content">Лабораторна робота №2(ч.2)</a></br>
<a href="sitelab1-2.html" target="content">Сайт-сторінка (ЛР 1-2)</a></br>
<a href="lab3.html" target="content">Лабораторна робота №3(ч.1)</a></br>
<a href="lab3.htm" target="content">Лабораторна робота №3(ч.2)</a></br>
<a href="lab3final_border2.html" target="content">Лабораторна робота №3(ч.3)</a></br>
<a href="anketa.htm" target="content">Лабораторна робота №4</a></br>
```

### 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 <b>circle, rect, poly</b>
alt	додає альтернативний текст підпису посилання для кожної області
coords	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">**

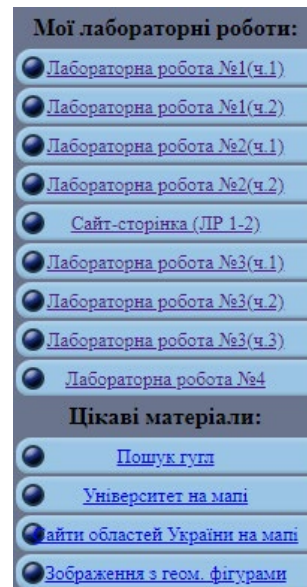
Для полігону задаються координати його вершин.

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

## 5.2 Практична частина

1. Запустіть веб-сервер Денвер.
2. Створіть файл **menu.html**. У файлі створіть меню наступним чином: розмістіть його у вигляді таблиці, у кожен рядок помістіть фоном фото з архіву до лабораторної роботи. Розбийте логічно файл на дві частини, перша – Ваші лабораторні роботи, а друга – інші елементи, створені під час лабораторної роботи.(див. рисунок справа). Крім фону у кожній клітинці створіть посилання зі звичайних лінків (тег «а») та обов'язково у тегі «а» вкажіть атрибут target, значенням якого має бути назва фрейму (атрибут name тегу iframe), який Ви будете в подальшому створювати під час цієї лабораторної роботи.
3. Створіть html-файл (lab6.html), що складається з двох фреймів.Для рівномірного розташування фреймів по горизонталі у лабораторній роботі пропонується, або скористатися розділенням сторінки на дві частини за допомогою таблиці, і вже у її клітинки розташовувати фрейми, або ж скористатися блочним елементом div, та вирівняти їх по горизонталі наступним чином: присвоїти їм атрибут class у який встановити будь-яке текстове значення латиницею. У «голові» документа, за допомогою тегу <Style> прописати наступні параметри:  

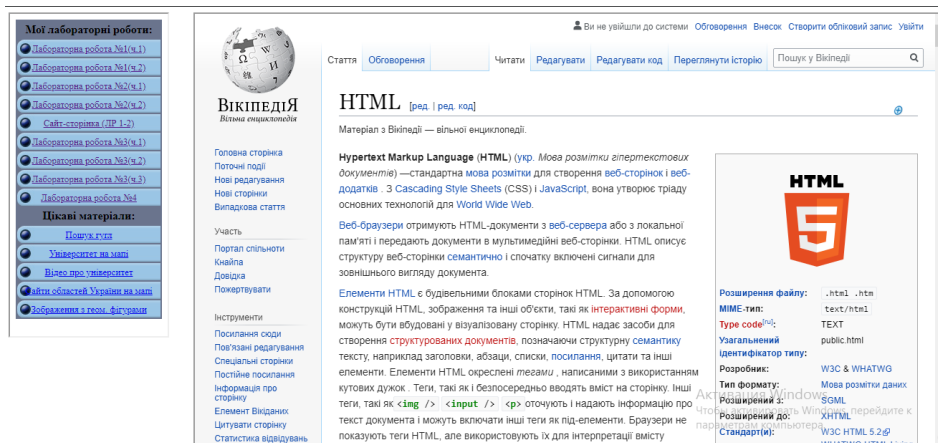
```
.*назва_класу_яку_ви_вказали {  
    display: inline;  
    vertical-align: top;  
}
```
4. Далі у файлі lab6.html у клітинках таблиці чи у «дівах» (відповідно до того, який метод Ви обрали) вставити теги <iframe>. Перший тег <iframe>має бути адресований на сторінку menu.html. Другий тег має за умовчанням завантажувати сторінку про HTML з української вікіпедії.
- 5.Розташуйте файл lab6.html на своєму веб-сервер і дайте кожному фрейму унікальне ім'я: значення атрибуту **name** для тега <iframe>. **Значення атрибуту name правого діва має співпадати зі значенням target, яке ми вказали у файлі menu.html.**
6. Задайте лівому фрейму фіксовану ширину в 220 пікселі.У лівий фрейм помістіть файл меню (**menu.html**).Зробіть так, щоб цей фрейм відображався як елемент основної сторінки.Рамку навколо нього – вимкніть.
7. Створіть в папці свого домену файл **map.html**, що містить карту зображень.
8. Помістіть у файл map.html зображення-карту з геометричними фігурами:  
``



9. 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>
```

10. Перевірте роботу карти зображень. Зверніть увагу на спливаючі підказки і посилання.
11. 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 .
12. Додайте у файл, що знаходиться в лівому фреймі посилання на обидві карти зображень.
13. Створіть пункт меню у файлі menu.html, який буде завантажувати сторінку академічного словника української мови <http://sum.in.ua>.
14. Створіть сторінку gmap.html. Напишіть у ній текст «Цю сторінку зробив студент групи \*назва групи\* \*Ваш ПІБ\* Національного Технічного Університету «Дніпровська Політехніка». Заняття з цієї дисципліни проходять за адресою вказаною на мапі». Після чого зайти на сайт GoogleMaps, знайти там корпус Вашого університету, де проходять заняття з цієї дисципліни. Натисніть на нього один раз лкм, щоб поставити маркер. Зліва з'явиться інформація про це місце. Натисніть у цій області на кнопку «поділитися», далі оберіть пункт «Вбудовування карт» та скопіюйте HTML код, який з'явиться на екрані, після чого встановіть його у цю сторінку. Змініть ширину мапи на 1000px, висоту залиште за замовченням.
15. Додайте лінки на сторінку тлумачного языка та gmap.htmlly menu.html
16. Створіть сторінку video.html. Перейдіть у Ютуб на офіційний канал НТУ Дніпровська Політехніка, або на канал кафедри КТЕД (<https://www.youtube.com/user/3НАНUE/videos>). Оберіть декілька відео, перейдіть на сторінку з ними, після чого натисніть «поділитися», після чого у вспливаючому вікні, оберіть «вбудувати», та скопіюйте лінк. Змініть ширину відео на 400px, а висоту на 285px.
17. Покажіть виконану роботу викладачеві. Кінцева робота має виглядати наступним чином:



## 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 використанню карт зображень?