МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Web-технології

Методичні вказівки до комп’ютерного практикуму

Для студентів напряму підготовки

6.050201 «Системна інженерія»

кафедри технічної кібернетики

всіх форм навчання

*Рекомендовано*

*Вченою радою факультету*

*інформатики та обчислювальної*

*техніки НТУУ «КПІ»*

*Протокол № \_ від \_\_\_\_\_2015 р.*

Київ

НТУУ «КПІ»

2015

Web-технології. Методичні вказівки до комп’ютерного практикуму. [Текст] / Уклад.: Є.В. Крилов– К.: НТУУ «КПІ», 2012. – 47 с.

Методичні вказівки призначені для студентів напряму підготовки 6.050201 «Системна інженерія» кафедри технічної кібернетики всіх форм навчання. В посібнику наведена тематика комп’ютерного практикуму, теоретичні відомості, завдання та послідовність виконання комп’ютерного практикуму, список літератури, контрольні питання.

Укладач Є.В.Крилов, к.т.н., доцент

Відповідальний редактор В.П. Пасько, к.т.н., доцент

Рецензент Новацький А.О, к.т.н.,

доцент кафедри АУТС

*За редакцією укладачів*

**Зміст**

[Вступ 5](#_Toc326661730)

[Мета, основні завдання та послідовність виконання робіт 6](#_Toc326661731)

[Обладнання для виконання комп’ютерного практикуму 7](#_Toc326661732)

[Заходи безпеки під час виконання комп’ютерного практикуму 7](#_Toc326661733)

[Лабораторна робота №1 8](#_Toc326661734)

[Вивчення HTML 8](#_Toc326661735)

[Методичні вказівки 8](#_Toc326661736)

["Створення HTML сторінок та їх оформлення" 8](#_Toc326661737)

[Лабораторна робота №2 13](#_Toc326661738)

[Вивчення CSS 13](#_Toc326661739)

[Завдання 1 13](#_Toc326661740)

[Завдання 2 14](#_Toc326661741)

[Завдання 3 15](#_Toc326661742)

[Завдання 4 16](#_Toc326661743)

[Завдання 5 17](#_Toc326661744)

[CSS методичні вказівки 18](#_Toc326661745)

[«Використання CSS для управління форматуванням web-сторінок» 18](#_Toc326661746)

[Лабораторна робота №3 20](#_Toc326661747)

[Вивчення розширеної мови розмітки тексту XML 20](#_Toc326661748)

[Теоретичні відомості: 20](#_Toc326661749)

[Питання для самоперевірки 21](#_Toc326661750)

[Лабораторна робота №4 22](#_Toc326661751)

[Технологія JavaScript. 22](#_Toc326661752)

[Теоретичні відомості: 22](#_Toc326661753)

[Включення скриптів JavaScript в HTML-сторінку 22](#_Toc326661754)

[**Інші приклади** 24](#_Toc326661755)

[Лабораторна робота №5 30](#_Toc326661756)

[Створення календаря подій на PHP. 30](#_Toc326661757)

[*Вивчення мови програмування РНР, робота з файлами.* 30](#_Toc326661758)

[Завдання: 30](#_Toc326661759)

[Теоретичні відомості: 31](#_Toc326661760)

[Інтеграція PHP і HTML 31](#_Toc326661761)

[*Приклад 4* 33](#_Toc326661762)

[*Приклад 5* 33](#_Toc326661763)

[*Приклад 6* 35](#_Toc326661764)

[*Приклад 9* 36](#_Toc326661765)

[*Приклад 12* 41](#_Toc326661766)

[*Приклад 13* 41](#_Toc326661767)

[*Приклад 14* 43](#_Toc326661768)

[Лабораторна робота №6 45](#_Toc326661769)

[Вивчення мови програмування РНР, роботи з функціями файлової системи. 45](#_Toc326661770)

[Завдання: 45](#_Toc326661771)

[Лабораторна робота №7 46](#_Toc326661772)

[Вивчення MySQL 46](#_Toc326661773)

[Завдання 46](#_Toc326661774)

[Теоретичні відомості: 47](#_Toc326661775)

# Вступ

Курс “Web-технології” відноситься до професіонально-орієнтованих дисциплін. Він базується на розділах дисциплін “Алгоритмічні мови програмування”, “Комп’ютерна графіка”. В свою чергу курс “Web-технології” орієнтований на спеціальні дисципліни програмування.

В курсі вивчається теоретична та інструментальна база, за домогою якої створюються інженерні методи вирішення проблеми розробки ефективного програмного забезпечення Інтернет-проектів. В курсі знайшли відображення сучасні інформаційні засоби створення Web-сайтів. Також подаються основи мови програмування PHP для розробки систем електронної комерції.

# Мета, основні завдання та послідовність виконання робіт

В ході навчання слухачі:

* вивчать принцип гіпертексту, основи мови HTML і специфікації CSS (каскадних таблиць стилів), достатні для створення веб-сторінок навіть в програмі Блокнот;
* отримають базові знання зі стилістики і структури Web, які допоможуть їм розробляти інформативні та зручні у використанні Web вузли;
* дізнаються всі етапи розробки веб-сайту, від планування до реклами;
* вивчать програму Macromedia Dreamweaver і з її допомогою створять свій власний сайт;
* розглянуть всі популярні принципи верстки сайтів - табличну, гібридний і фіксований дизайни, фрейми та ін.;
* дізнаються, як і де розмістити розроблені сайти, розглянуть особливості комерційного та безкоштовного хостингу;
* вивчать вимоги до web-графіки та текстової інформації;
* освоять основні прийоми, які використовуються при розробці макета web-ресурсу;
* вивчать методи підготовки статичних і динамічних елементів Інтернет-сторінок, побудови систем навігації;
* навчаться створювати анімацію, анімовані банери, ознайомляться з видами реклами в Інтернет;
* ознайомляться з сучасними методами планування та розробки web-ресурсу;
* навчаться створювати динамічні елементи;
* вивчать сучасні технології, які використовуються при верстці сайтів і зверстати сайти декількома способами;

самостійно створять векторні піктограми, макети web-сторінок

Виконання лабораторних робіт по курсу WEB-технології відбувається наступним чином:

* Необхідно завантажити на робочій станції операційну систему Windows XP for student (друга в списку) і зайти з логіном student без пароля.
* Запустити програму Internet Explorer.
* В адресному рядку набрати Server436.ua і вибрати розділ WEB-технології в головному меню.
* Лабораторні роботи 1(HTML), 2(CSS), 3(XML), 4(Java Script) виконуються на робочих станціях (не вимагають веб-сервера), а потім для здачі копіюються на сервер в розшарену папку\\server436\HTML\ де необхідно створити свою папку в папці з відповідним номером групи.
* Лабораторні роботи 5(PHP), 6(PHP), 7(My SQL) виконуються на робочих станціях, а потім для здачі копіюються на сервер в розшарену папку \\server436\PHP\ де необхідно створити свою папку в папці з відповідним номером групи.
* Для виконання \*.PHP файлів необхідно в адресному рядку браузера (Internet Explorer) набрати http://<номер групи>.student/ (наприклад http://ik51.student/) і запустити свої файли.

# Обладнання для виконання комп’ютерного практикуму

Лабораторні роботи виконуються в комп’ютерному класі, обладнаному сервером та робочими станціями, які об’єднанні в локальну мережу. На головному комп’ютері повинен бути розгорнутий Web-сервер.

# Заходи безпеки під час виконання комп’ютерного практикуму

Всі студенти повинні дотримуватись правил пожарної безпеки та правил техніки безпеки при роботі в кабінетах інформатики кафедри технічної кібернетики НТУУ «КПІ», затверджений завідуючим кафедрою. На першому занятті необхідно провести інструктаж. Кожний студент повинен поставити свій підпис в журналі інструктажу з техніки безпеки.

# Лабораторна робота №1

## Вивчення HTML

**Мета роботи:** Навчитися створювати статичні веб сторінки.

Теоретичні відомості:

1) Основні теги HLML:

1.1 Теги області HEAD(TITLE, META, LINK, BASE).  
1.2 Теги вирівнювання (H ‹номер›, BR, DIV, FONT, P ).  
1.3 Теги списків (OL, UL, MENU, DL, LI, DT).  
1.4 Теги ссилок А(параметри name, href, title, mailto ).  
1.5 Теги таблиць (table, caption, tr, td, th, параметри rowspan, colspan).  
1.6 Теги відображення зображень IMG.

2) Робота з редактором Dream Weaver електронний підручник.  
3) Основи оптимізації сайтів (контекстна оптимізація, ссилочна оптимізація ).

**Завдання:** В даній лабораторній роботе студенту необхідно створити статичну web-сторінку на одну з наступних тем:

* + Продаж автомобілів.
  + Виставка медичного устаткування.
  + Подорожі.
  + Моя група.
  + Тема по вибору студента.

В HTML коді створеної web-сторінки обов’язково мають бути всі теги вказані в п.1 теоретичних відомостей.

## Методичні вказівки

## **"Створення HTML сторінок та їх оформлення"**

**Для створення звичайного тексту використовуються наступні теги:**

<HTML>  
<HEAD>  
<TITLE>Мій перший документ на HTML</TITLE>  
</HEAD>  
<BODY>  
<P>Всім привіт!  
</BODY>  
</HTML>

**Для більш читабельного оформлення тексту використовують параметр "рівень заголовка"**

<HTML>  
<HEAD>  
<TITLE>Форматування</TITLE>  
</HEAD>  
<BODY>  
<H1>Заголовок 1-го рівня</H1>  
<H2>Заголовок 2-го рівня</H2>  
<H3>Заголовок 3-го рівня</H3>  
<H4>Заголовок 4-го рівня</H4>  
<H5>Заголовок 5-го рівня</H5>  
<H6>Заголовок 6-го рівня</H6>  
</BODY>  
</HTML>

**Гіпертекстовий документ може бути оформлений з використанням наступних стилів:**

<b>напівжирний</b>  
<i>курсив</i>  
<cite>цитата</cite>  
<code>програмний код</code>  
<dfn>визначення</dfn>  
<em>логічний акцент</em>  
<kbd>ввід з клавіатури</kbd>  
<samp>повідомлення комп’ютера</samp>  
<strong>сильний акцент</strong>  
<var>змінні</var>

**Зміна шрифта тексту:**

<p><font face="Arial">Текст Arial</font></p>  
<p><font face="Times New Roman">Текст Times New Roman</font></p>  
<p><font face="Courier New, Courier, mono">Текст Courier New</font></p>.

**Вирівнювання тексту може бути:**

<p align="left">лівим</p>  
<p align="right">правим</p>  
<p align="center">по центру</p>

**Види списків:**

<H1>Види списків</H1>

звичайний:  
<ul>  
<li>Елемент списку  
<li>Ще один елемент списку  
</ul>

нумерований:

<ol>  
<li> Елемент списку  
<li> Ще один елемент списку  
</ol>

c описом:

<DL>  
<DT> Собака (елемент)  
<DD> Друг людини (опис елементу)  
<DT> Кіт (елемент)  
<DD> Домашня тварина (опис елемента)  
</DL>

вкладені:  
<UL>  
<LI>Примус  
<OL>  
<LI>інший примус  
...  
</OL>  
<LI>  
...  
</UL>

**Для зміни кольору тексту і фону використовується наступна форма:**

<body bgcolor="#CCFFCC" text="#666600">  
<h1>Зміна кольору тексту і гіперпосилань</h1>  
<p>Всі<font color="red">питання</font>, <font color="green">зв’язані</font>з<font color="blue">реєстрацією</font>

**Типи посилань:**

<p>Посилання на **<a href="href1.htm">**web-сторінку</a></p>  
<p> Посилання на адрес електронної пошти: <**a href="mailto:**dr\_jack@ukr.net">dr\_jack@ukr.net</a></p>  
<p> Посилання на FTP-сервер: **<a href="ftp:/**/ftp.microsoft.com">ftp.microsoft.com</a></p>  
<p> Посилання на групу новин**<a href="**news:rec.arts.books">rec.arts.books</a></p>

**Використання таблиць:**

**Звичайна таблиця**  
<table border="1">  
<tr>  
<td>Комірка 1</td>  
<td>Комірка 2</td>  
<td>Комірка 3</td>  
</tr>  
<tr>  
<td>Комірка 4</td>  
<td>Комірка 5</td>  
<td>Комірка 6</td>  
</tr>  
<tr>  
<td>Комірка 7</td>  
<td>Комірка 8</td>  
<td>Комірка 9</td>  
</tr>  
</table>

**Використання rowspan**   
<table border="1">  
<tr>  
<td rowspan="2">Комірка 1- 4</td>  
<td>Комірка 2</td>  
<td>Комірка 3</td>  
</tr>  
<tr>  
<td>Комірка 5</td>  
<td>Комірка 6</td>  
</tr>  
<tr>  
<td>Комірка 7</td>  
<td>Комірка 8</td>  
<td>Комірка 9</td>  
</tr>  
</table>

**Використання colspan**   
<table border="1">  
<tr>  
<td colspan="2">Комірка 1- 2</td>  
<td>Комірка 3</td>  
</tr>  
<tr>  
<td>Комірка 4</td>  
<td>Комірка 5</td>  
<td>Комірка 6</td>  
</tr>  
<tr>  
<td>Комірка 7</td>  
<td>Комірка 8</td>  
<td>Комірка 9</td>  
</tr>  
</table>  
  
**Вкладені таблиці**  
<table border="1">  
<tr>  
<td>  
<table border="1">  
<tr>  
<td>Комірка 1</td>  
<td>Комірка 2</td>  
<td>Комірка 3</td>  
</tr>  
<tr>  
<td>Комірка 4</td>  
<td>Комірка 5</td>  
<td>Комірка 6</td>  
</tr>  
<tr>  
<td>Комірка 7</td>  
<td>Комірка 8</td>  
<td>Комірка 9</td>  
</tr>  
</table>

# Лабораторна робота №2

## Вивчення CSS

**Мета роботи:**Вивчення можливості оформлення web-сторінок з використанням каскадних стилів CSS.

**Теоретичні відомості:**

1) Підключення CSS (теги META, LINK);

2) Впровадження CSS (тег STYLE);

3) Вбудований CSS (параметр STYLE);

4) Імпорт  CSS (import);

5) Конкуруючі CSS (автор, користувач, браузер; оперативні, впровадження, зв’язані; параметр importcnt) ;

6) Правила CSS (H1, P, UL, LI и др.);

7) Наслідування CSS (OL LI; UL LI;...);

8) Класи CSS;

9) Псевдо класи CSS.

**Завдання:** В даній лабораторій роботе студент використовує web-сторінку створену ним в лабораторній роботі №1.При цьому він повинен створити в окремому файлі опис каскадних стилів і зв’язати тегом LINK файл .CSS с файлом .HTML.

### Завдання 1

1. Відкрийте будь-який раніше створений вами HTML-документ в режимі редактора Блокнот. Встановіть для кількох абзаців стиль Заголовок 1, збережіть документ і перегляньте його у вікні браузера.
2. Знову відкрийте даний документ в режимі перегляду HTML-коду і встановіть для одного із заголовків атрибут ‹style›: ‹h1 style = "font-style: italic; color: navy"›. Мій новий стиль заголовка ‹/h1›.
3. Застосуйте загальний стиль для заголовка і абзацу використавши тег <div>: вирівнювання по центру, шрифт - підкреслений.

***‹ div style = "text-align: center; text-decoration: underline "›***

***‹h1› Наглядне відображення ‹/h1›***

***‹p› використання тега div‹/P›***

***‹h1 style = "color: red; font-family: courer "›‹span style = "color: Green "›для тексту ‹/span›, заданого стилем‹h1›***

***‹h1 style = " font- style: italic; color: nevy"› Ще один ‹span style = "color: Green ""› стиль ‹/span› тексту. ‹h1›"***

***‹/div›***

Збережіть зміни і перегляньте документ у вікні браузера.

Створення html-документа з вбудованою таблицею стилів виконується за допомогою парного тегу <style>, який слід розмістити в head-блоці.

**Синтаксис тегу <style>:**<style disabled media=screen | print | all titlt = text type = "text/css">

Type - єдиний обов'язковий атрибут тегу <style>. Цей атрибут задає тип mime (Multipurpose Іnternet Mail Extension, стандарт електронної пошти Інтернет). Для того щоб браузери, які не підтримують вбудовані таблиці стилів, мали можливість їх ігнорувати, атрибуту присвоюється значення text / css.

Інструкції стилів набираються всередині тегів коментарів для того, щоб браузери, які не розпізнають таблиці стилів, не виводили вміст тегу <STYLE> в web-сторінку. Коментарі всередині таблиці стилів розташовуються між символами / \* і \* /, тому ігноруються браузером.

### Завдання 2

Відкрийте будь-який раніше створений вами html-документ в режимі редактора Блокнот.

В область head даного документа введіть коди таблиці стилів.

***‹html›  
  
‹head›  
  
‹ style type = “text/css”›  
  
‹! --***

***body {background: yellow; color: green}***

***/\* Жовтий фон документу з зеленими літерами \*/  
  
h1 {font: 20pt Arial bold; color: red}***

***/\* Заголовок h1:напівжирний шрифт Arial висотою 20 червоного кольору \*/  
  
р {font: 12pt Arial; text-indent: 1.25cm}***

***/\* Параграф: червоний рядок в 1.25 см, шрифт Arial висотою 12 \*/  
  
--›  
  
‹/style›  
  
‹/head›  
  
‹body›  
  
‹h1› Новий стиль заголовка ‹/h1›  
  
‹р›  
  
Новий стиль параграфа.  
  
‹/р›  
  
‹/body›  
  
‹/html›***  
  
Збережіть документ.

Скопіюйте таблицю стилів в буфер обміну і вставте її в інший html-документ.

Таблицю стилів можна створити на основі окремого СSS-файла, з наступним застосуванням його до документів. Таблиця стилів зберігається в текстовому СSS-файлі.

Для зв'язування використовується тег <link>, який повинен розташовуватися між парними тегами <HEAD>.

Файл-СSS з визначенням стилю можна розмістити на сервері і потім додавати в заголовок кожного документу наступний фрагмент:

<Link Rel="Stylesheet" Type= "Text/сss "src= [http://ім’я\_серверу/ім’я\_файла/сss](http://ім'я_серверу/ім'я_файла/сss)>

Якщо документ, до якого будуть застосовуватися стилі, знаходиться в одному каталозі з css-файлом, тоді просто вказується посилання на стильовий файл:

<Link Rel = "Stylesheet" href = "/ NewStyle.css" type = "text / css">

При завантаженні сторінки браузер буде завантажувати також стилі, зазначені в елементі LINK. Тому при внесенні змін до css-файлу, зміниться і зовнішній вигляд сторінок всього сайту.

### Завдання 3

Створіть CSS файл і застосуйте його для форматування раніше створених вами сторінок.

Для створення стильового файлу Вам будуть потрібні кілька графічних файлів: фонове зображення, маркери, лінії та інші (файли \*.jpg або \*.gif).

Завантажте програму Блокнот і впишіть наступний код:  
  
***Body{font-family: Verdana, Arial, Helvetica, sans-serif; color: rgb(0,90,90);***

***font-size: 12px; background-image: url('/\*.jpg');}***

***P{ font-family: Verdana, Arial, Helvetica, sans-serif; color: rgb(0,90,90);}***

***u1{list-style-image: url('/\*.gif');}***

***a {color: #707986; text-decoration: none;}***

Збережіть документ, як файл stylesl.css (при збереженні документа встановіть в поле «Тип файлу» категорію «Усе».

Завантажте 2-3 раніше створених html-документів. В область head кожного документу вставте посилання на стильовий файл:

<link Rel="stylesheet" href="/ stylesl.css "type= "text/сss ">

Збережіть документи і перегляньте їх у вікні браузера.

Внесіть зміни в стильовий файл stylesl.css. Наприклад, замініть файл фонового зображення і збережіть документ. Перейдіть в режим перегляду документа в браузері і виконайте команду «Оновити сторінку».

Створіть новий стильовий файл styles2.css (наприклад, замініть у файлі stylesl.css файли фонового зображення, маркерів, стилі заголовків, абзаців і т.д. і збережіть його під іншим ім'ям).

Відкрийте кілька html-документів в режимі html-коду і замініть ім'я стильового файлу в записі посилання.

***‹link rel="stylesheet" href="/ styles2.css " type= "text/сss "›***

Збережіть зміни і перегляньте результат у вікні браузера.

Додаткову гнучкість застосування таблиці стилів дає можливість визначення в ній різних класів стилів для одного і того ж елемента, які потім можна застосовувати до елементів документу.

***Синтаксис запису класу:*** .Ім’я Класу {властивість: значення; властивість: значення; ...}

### Завдання 4

Загрузіть стильовий файл stylesl.css і доповніть його кодами класів стилів для абзацу, заголовка и гіперпосилання:  
  
***Body {font-family: Verdana, Arial, Helvetica, sans-serif; color: rgb(0,90,90);***

***font-size: 12px; background-image: url('/\*0.jpg') ;}***

***P{ font-family: Verdana, Arial, Helvetica, sans-serif; color: rgb(0,90,90);}***

***.fon{ background-image=url('/\*1.jpg') ;}***

***.fon1{text-align: center; background-image=url('/\*2.gif') ; }***

***u1{list-style-image: url('/\*1.gif');}***

***.ul1{list-style-image: url('/\*2.gif');}***

***.ul2{list-style-image: url('/\*3.gif');}***

***.ul3{list-style-image: url('/\*4.gif');}***

***.ul4{list-style-image: url('/\*5.gif');}***

***H1.Border {border-width: 1; border: solid; border-color: green; text-align:center}***

***H1.Background {background-color: lime; text-align: center}***

***a {color: #707986; text-decoration: none;}***

***.prod {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;***

***color: #2D3368; font-style: normal; font-weight: bold; text-decoration:none; line-height: 15px;}***

***.prod:hover {font-family: Verdana, Arial, Helvetica, sans-serif; font-size:10px;***

***color: #A57329; font-style: normal; font-weight: bold; text-decoration: none; line-height: 15px; }***

***.menuurl {font-family: Verdana, Arial, Helvetica, sans-serif; color:#222222;***

***font-size: 20px; font-style: normal; font-weight: bold; }***

Збережіть зміни в файлі stylesl.css.

### Завдання 5

Загрузіть створений раніше html-документ і визначте для окремих елементів документу класи стилів, наприклад:

***html›‹head›***

***‹link href="/stylesl.css" mce\_href="/stylesl.css" rel="stylesheet" type="text/css"›***

***‹/head›***

***‹body›  
‹р›Приклад тексту‹/р›***

***‹p class ="fon"›Приклад тексту‹/р›***

***‹H1 class="Border"›Заголовок першого рівня з рамкою‹/Н1›***

***"‹Н1 class= "Background"›Заголовок першого рівня з фоном‹/Н1›***

***‹Н1›Заголовок першого рівня стандартний‹/Н1›***

***‹/body›‹/html›***

Збережіть документ і перегляньте його у вікні браузера.

## CSS методичні вказівки

## «Використання CSS для управління форматуванням web-сторінок»

CSS, Cascading Style Sheet - каскадні таблиці стилів дозволяють web-розробнику відокремити процес форматування сторінки від заповнення її конкретним змістом і дають можливість визначити форму подання інформації від її фактичного змісту, тобто можливість підготувати документ, а потім незалежно редагувати використовувані в ньому стилі.

CSS складаються з одного або декількох правил, що визначають особливості відображення елементів web-сторінки.

Синтаксис правила:

***Селектор {властивість: значення}.***

Селектор - позначення дескриптора, до якого застосовується правило, а властивість - атрибут.

Можна поставити кілька властивостей, відокремивши їх один від одного крапкою з комою.

***Приклади:***

1. **H1 {color: red}** - зміна дії елемента HI. Встановлюється червоний колір тексту для усіх тегів HI. Таким чином, можна змінювати параметри тексту в кожному блоці. Параметр color, має назву властивість, a red-значенням властивості.

2. **H1 {font: 16pt "" Arial ""; font-weight: bold; margin-left: 0.25in; color: black}** - встановлена ​​насиченість шрифту, межі і колір тексту.

***Правила при роботі з CSS:***

Коментарі можуть бути присутніми в будь-якій позиції і починатися символами / \* і закінчуватися \* /.

Правила CSS не чутливі до регістру символів, виключення значення безпосередньо пов'язані з CSS наприклад, імена шрифтів, URL, HTML-атрибути і т.д.).

Імена селекторів можуть включати літери (А / а - Z / z), цифри 0-9 і знаки підкреслення, але перед першим символом повинна стояти буква.

***Розподіл стилів за способом застосування:***

1. **Inline Style** - завдання стилів для окремого фрагменту документа;

2. **Embedding Style** - вбудовування таблиці стилів в документ за допомогою парного тегу;

3. **Linking Style** - зв'язування документа з таблицею стилів, яка зберігається в окремому файлі (html-файл, простой текстовий файл, але з розширенням СSS, що дозволяє централізовано керувати стилями).

Завдання стилів для окремого фрагменту документа виконується встановленням значення атрибуту тегу цього фрагмента.

***Синтаксис:***<tag style = "attribute: value; attribute: value;... "></ tag>

Можливе використання атрибута style для тегу <div> ..... </ div> дозволяє керувати стилем фрагмента, розміщеного між кордонами парного тега <div>

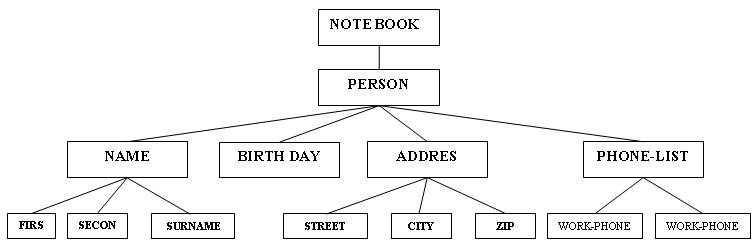
# ****Лабораторна робота №3****

## Вивчення розширеної мови розмітки тексту XML

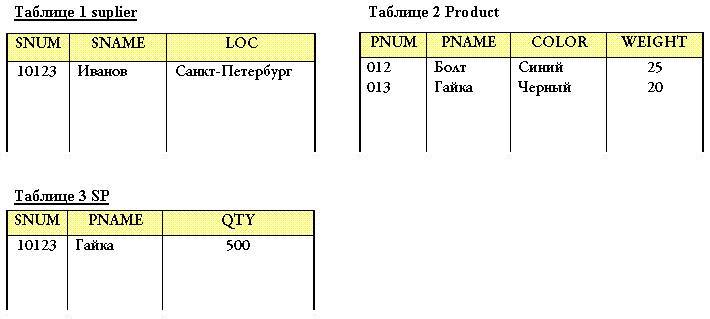
### Теоретичні відомості:

**Завдання:**

Розробити добре оформлений XML документ наступної структури:



1. Подати тег name у вигляді атрибутів first, second, surname.
2. Застосувати для всіх імен XML документу префікс простору імен.
3. Записати у вигляді XML документу вибірку з БД " ".



### ****Питання для самоперевірки****

**Питання:**

1. Що означає «добре оформлений» документ?
2. Що означає "правильний" документ?
3. Чи обов’язковий пролог документу?
4. Знайдіть помилки в записі ‹support id = 1234 › sd › 12xz ‹/support›?
5. Чи правильний запис ‹item selected› сірники ‹/item›?
6. Якими способами можна записати символ "менше" ?
7. Чи можна використовувати однакові імена у вкладених елемента, якщо дані імена вкладені в різні елементи?
8. Чи можна записати атрибути у закриваючому тезі?
9. Як можна запобігти розбору елементу програмою-аналізатором?

# Лабораторна робота №4

## Технологія JavaScript.

**Завдання:** Освоїти навички програмування в JavaScript. Створити набір логічно пов'язаних html-сторінок з включеннями JavaScript. Необхідно самостійно, на основі прикладів, розробити оригінальні операторні конструкції з використанням обробників подій, функції з використанням об'єктів JavaScript.

### Теоретичні відомості:

***Коментарі***

### Включення скриптів JavaScript в HTML-сторінку

Текст програми на мові JavaScript вставляється безпосередньо в файл HTML-сторінки. Щоб подивитися, як це виглядає, наведемо приклад найпростішої HTML-сторінки.

<html>

<body>

<br>

Це звичайний HTML документ

<br>

<script language="JavaScript">

document, write ("Це і є JavaScript!")

</ Script>

<br>

Виходимо назад в HTML

</ Body>

</ Html>

На перший погляд все виглядає, як звичайний HTML-файл. Єдиною відмінністю є те, що в ньому міститься наступний фрагмент:

<script language="JavaScript">

document.write ("Це і є JavaScript!")

</ Script>

Цей фрагмент являє собою скрипт, написаний на мові JavaScript.

Щоб подивитися, як такий скрипт буде працювати, наберіть текст HTML-сторінки, наведений вище. Потім збережіть його, як звичайну HTML-сторінку і завантажте в браузер з активованим JavaScript.

Звичайно, цей скрипт навряд чи може виявитися корисним, адже подібну інформацію можна записати за допомогою звичайного HTML набагато простішим способом. Метою цього прикладу було показати, як вставляється JavaScript в файл HTML-документа і продемонструвати призначення нових ярликів <script> і </script>. Вся інформація, яка розташовується між цими тегами сприймається як програма на мові JavaScript. У тексті програми зустрічається рядок document.write() - одна з найбільш важливих команд в програмуванні на JavaScript. Рядок document.write() використовується тоді, коли потрібно записати щось в документ, в даному випадку, в HTML-документ.

***Події***

Подія - це дуже важливе в програмуванні на JavaScript поняття. Події головним чином породжуються користувачем, є наслідками його дій. Якщо користувач натискає кнопку миші, то відбувається подія, яка має назву Click. Якщо екранний покажчик миші рухається за посиланням HTML-документа, відбувається подія Mouse Over.

Існує кілька різних подій. Нехай ми хочемо, щоб наша JavaScript програма реагувала на кілька різних подій. Це можна зробити за допомогою event-handlers. Наприклад, можна змусити з'явитися нове спливаюче вікно, яке з'являється при натисканні кнопки. Поява нового вікна буде наслідком настання події Click.

Event-handler, інакше кажучи обробник подій чи засіб управління подіями, який потрібний для організації такої дії, називається onClick. Цей засіб управління подіями повідомляє комп'ютеру, які дії необхідно зробити при настанні даної події. Нижче показана проста програма, яка ілюструє те, як може бути використано засіб on Click.

***Текст скрипта:***

<form>

<input type="button" value="Click me" onClick="alert('Ой-ой')">

</ Form>

У наведеному вище прикладі є кілька нових моментів. Розглянемо все докладно, крок за кроком. У цьому прикладі ми описали форму або бланк, в якому є кнопка. Це здійснюється за допомогою інструкцій HTML, і тому ми не будемо на цьому зупинятися. Для нас важливою частиною є інструкція onClick = "alert ('Ой-ой')", яка розміщена всередині ярлика <input>.

Вона визначає ту дію, яку буде виконано при натисканні кнопки. Якщо відбувається подія Click, то комп'ютер виконає інструкцію аlert ('Ой-ой'). Ця інструкція - фрагмент JavaScript. Зауважте, що в цьому випадку ми не використовували ярлик <script>.

Інструкція alert() дозволяє створювати спливаючі вікна. В дужках слід записати рядок, в нашому випадку це 'Ой-ой’. Текст рядка буде видно у спливаючому вікні. Наш скрипт змушує з'явитися на екрані нове вікно з текстом 'Ой-ой', коли користувач натискає на кнопку.

Відзначимо маленьку деталь, яка, однак, може створити деякі непорозуміння. У команді document.write() ми писали подвійні лапки ", а в комбінації інструкцій, що містить alert(), ми використовуємо одиночні лапки '- чому? Взагалі кажучи, можна користуватися як тими, так і іншими лапками.

У нашому останньому прикладі ми використовуємо дві пари лапок і пишемо оnСlick = "аlert('Ой-ой')". Але якщо ми напишемо оnСlick = "аlert (" Ой-ой ")", то комп'ютер не зрозуміє, яка частина належить до onClick, а яка - ні.

***Приклад функції з поєднанням засобів управління подіями***

***Текст скрипта:***

***‹html›***

***‹head›***

***‹script language="JavaScript"›***

***‹!-- hide***

***function calculation()***

***{***

***var x= 12;***

***var y= 5;***

***var result= x + y;***

***alert(result);***

***}***

***// --›***

***‹/script›***

***‹/head›***

***‹body›***

***‹form›***

***‹input type="button" value="Calculate" onClick="calculation()"›***

***‹/form›***

***‹/body›***

***‹/html›***

**Інші приклади**

*‹form name="myForm"› ‹input type="text" name="input" value="Ха-ха-ха"› ‹input*

*type="button" value="Пишемо свій текст (Ой!)"*

*onClick="document.myForm.input.value= 'Ой!'; "›*

*‹script language="JavaScript"›*

*function first() {*

*alert("The value of the textelement is: "+ document.myForm.myText.value);*

*}*

*function second()*

*{*

*var myString= "Поле для галочки ";*

*if (document.myForm.myCheckbox.checked) myString+= "відмічено" else*

*myString+= "не відмічено";*

*alert(myString); }*

*‹/script›*

*‹/head›*

*‹body bgcolor=lightblue›*

*‹form name="myForm"›*

*‹input type="text" name="myText" value="Ха-ха-ха"›*

*‹input type="button" name="button1" value="Кнопка 1" OnClick="first()"›*

*‹br›*

*‹input type="checkbox" name="myCheckbox" CHECKED›*

*‹input type="button" name="button2" value="Кнопка 2" onClick="second()"›*

*‹/form›*

*‹p›‹br›‹br›*

*‹script language="JavaScript"›*

*document.write("Колір фона: ");*

*document.write(document.bgColor + "‹br›");*

*document.write("Текст на другій кнопці: ");*

*document.write(document.myForm.button2.value);*

*‹/script›*

*‹script language="JavaScript"›*

*‹!—hide*

*function statbar(txt) {*

*window.status = txt;*

*}*

*// --›*

*‹/script›*

*‹/head›*

*‹body›*

*‹form›*

*‹input type="button" name="look" value=Пишемо на панель стану!"*

*onClick="statbar('Ось вона, панель стану!');"›*

*‹input type="button" name="erase" value=Стираємо текст з панелі стану!"*

*onClick="statbar('');"›*

*‹/form›*

*‹script language="JavaScript"›*

*‹!—hide*

*function timer() {*

*setTimeout("alert('Час вийшов! Вспливаю!')", 4000);*

*}*

*// --›*

*‹/script›*

*...*

*‹form›*

*‹input type="button" value="Часова затримка" onClick="timer()"›*

*‹/form›*

*‹script language="JavaScript"›*

*‹!—hide*

*// Задаєм вивідный текст*

*var scrtxt = "Цей текст виводиться з допомогою JavaScript! " +*

*"Цей текст виводиться з допомогою JavaScript! " +*

*"Цей текст виводиться з допомогою JavaScript!";*

*var length = scrtxt.length;*

*var width = 100;*

*var pos = -(width + 2);*

*function scroll() {*

*// виводимо текст з правої позиції і ставимо часову затримку*

*// зсуваємо позицію на один крок*

*pos++;*

*// визначаємо, яка частина тексту буде видима*

*var scroller = "";*

*if (pos == length) {*

*pos = -(width + 2);*

*}*

*//якщо текст не досяг лівого краю,*

*// то необхідно добавити декілька пробілів - інакше прийдеться*

*// відрізати передню частину тексту*

*if (pos ‹ 0) {*

*for (var i = 1; i ‹= Math.abs(pos); i++) {*

*scroller = scroller + " ";}*

*scroller = scroller + scrtxt.substring(0, width - i + 1);*

*}*

*else {*

*scroller = scroller + scrtxt.substring(pos, width + pos);*

*}*

*// виводимо текст на панель стану*

*window.status = scroller;*

*// знову звертаємось до функції після 100 мілісекунд*

*setTimeout("scroll()", 100);*

*}*

*// --›*

*‹/script›*

*‹/head›*

*‹body onLoad="scroll()"›*

*Тут розміщується текст HTML-сторінки.*

*‹/body›*

*‹/html›*

*‹script language="JavaScript"›*

*function test1(form)*

*{ if (form.text1.value == "")*

*alert("Please enter a string!")*

*else*

*{*

*alert("Hi "+form.text1.value+"! Form input ok!");*

*}*

*}*

*function test2(form)*

*{*

*if (form.text2.value == "" || form.text2.value.indexOf('@', 0) == -1)*

*alert("No valid e-mail address!");*

*else*

*alert("OK!"); }*

*‹/script›*

*‹body›*

*‹form name="first"› Enter your name:‹br› ‹input type="text" name="text1"›*

*‹input type="button" name="button1" value="Перевіримо ввід"*

*onClick="test1(this.form)"› ‹P› Enter your e-mail address:‹br› ‹input type="text"*

*name="text2"›*

*‹input type="button" name="button2" value="Перевіримо ввід"*

*onClick="test2(this.form)"› ‹/form›*

*function setfocus()*

*{*

*document.first.text1.focus();*

*}*

*function setfocus() {*

*document.first.text1.focus();*

*document.first.text1.select();*

*}*

*‹a href="#" onMouseOver="document.myImage2.src='img2.gif'"*

*onMouseOut="document.myImage2.src='img1.gif'"› ‹img src="/img1.gif"*

*name="myImage2" width=160 height=50 border=0›‹/a›*

# Лабораторна робота №5

## Створення календаря подій на PHP.

***Мета роботи:****Навчитися використовувати функції роботи з рядковими змінними і роботи з файлами для написання програм на PHP.*

*Теоретичні відомості:*

*1 Інтеграція PHP і HTML (теги ?Php, ?, Script,%).*

*2 Оператори і керуючі конструкції PHP (if, elseif, while, for, foreach, switch, break, continue).*

*3 Функції роботи з масивами (array, in-array, array-pop, expload)*

*4 Функції роботи з файлами (file\_exist, is\_file, fopen, fclose, fwrite, is\_writable, fread, is\_readable).*

### *Вивчення мови програмування РНР, робота з файлами.*

### Завдання:

*У даній лабораторній роботі студент повинен написати програму, яка відкриває текстовий файл, зчитує дані і виводить інформацію в браузері.*

*1)Створити текстовий файл.*

*Вихідний текстовий файл має наступний формат:*

*21.01.01 / 12:00 / підготовка до екзамену*

*22.01.01 / 14:00 / здача екзамену*

*23.01.01 / 15:00 / лекція*

*.....*

*.....*

*.....*

##### *2) Відкрити файл, зчитати дані.*

***Дані повинні відображатися в браузері в наступному форматі:***

*Календар подій*

|  |  |  |
| --- | --- | --- |
| *дата* | *час* | *подія* |
| *21.01.01* | *12:00* | *Підготовка до екзамену* |
| *22.01.01* | *14:00* | *здача екзамену* |
| *23.01.01* | *15:00* | *лекція* |
| *........* | *...* | *............* |

*3) При написанні програми необхідно врахувати наступні виключення:*

* *Текстовий файл може бути відсутнім*
* *Можливі помилки оператора при формуванні текстового файлу*

### Теоретичні відомості:

# Інтеграція PHP і HTML

*Приклад 1*

*<?*

*// Присвоїти значення декільком змінним*

*$site\_title = "PHP Recipes";*

*$bg\_color = "white";*

*$user\_name = "Chef Luigi";*

*?>*

*<html>*

*<head>*

*<title><? print $site\_title; ?></title>*

*</head>*

*<body bgcolor="<? print $bg\_color; ?>" >*

*<?*

*// Вивести привітання з датою і ім’ям користувача.*

*print "*

*PHP Recipes | ".date("F d, Y")." <br>*

*Greetings, $user\_name! <br>*

*";*

*?>*

*</body>*

*</html>*

*Приклад 2*

*<html>*

*<head>*

*<title>Basic PHP/HTML integration</title>*

*</head>*

*<body>*

*<?*

*// Зверніть увагу на присутність тегів HTML в команді print.*

*print "<h3>PHP/HTML integration is cool.</h3>";*

*?>*

*</body>*

*</html>*

*Приклад 3*

*<html>*

*<head>*

*<title>PHP Recipes | <? print (date("F d, Y")); ?></title>*

*</head>*

*<?*

*$big\_font = "h3";*

*?>*

*<body>*

*<? print "<$big\_font>PHP Recipes</$big\_font>"; ?>*

*</body>*

*</html>*

### *Приклад 4*

*<? require ('init.tpl'); ?>*

*<html>*

*<head>*

*<title><? print $site\_title; ?></title>*

*</head>*

*<body>*

*<? print "Welcome to $site\_title. For questions, contact <a href =*

*\"mailto:$contact\_email\">$contact\_name</a>."; ?>*

*</body>*

*</html>*

### *Приклад 5*

*?*

*// Файл: header.tpl*

*// Призначення:Головний файл для сайту PhpRecipes*

*// Дата: 22серпня 2000 г.*

*$site\_name = "PHPRecipes";*

*$site\_email = "wjgilmore@hotmail.com";*

*$site\_path = "http://localhost/phprecipes";*

*?>*

*<html>*

*<head>*

*<title><? print $site\_name; ?></title>*

*</head>*

*<body bgcolor="#7b8079" text="#ffffff" link="#e7d387" alink="#e7d387" vlink="#e7f0e4">*

*<table width = "95%" cellpadding="0" cellspacing="0" border="1">*

*<tr>*

*<td valign = "top">*

*PHPRecipes*

*</td>*

*<td valign = "top" align="right">*

*<?*

*// Вивести поточну дату і час*

*print date ("F d, h:i a");*

*?>*

*</td>*

*</tr>*

*</table>*

### *Приклад 6*

*<table width="95%" cellspacing="0" cellpadding="0" border="1">*

*<tr><td valign="top" align="middle">*

*Copyright &copy; 2000 PHPRecipes. All rights reserved.<br>*

*<a href = "mailto:<?=$site\_email;?>">contact</a> | <a href = "<?=$site\_path;?>/privacy.php">your*

*privacy</a>*

*</td></tr>*

*</table>*

*</body>*

*</html>*

*Приклад 7*

*<table width="95%" cellspacing="0" cellpadding="0" border="1">*

*<tr>*

*<td valign="top" width="25%">*

*<a href = "<?=$site\_path;?>/tutorials.php">tutorials</a><br>*

*<a href = "<?=$site\_path;?>/articles.php">articles</a><br>*

*<a href = "<?=$site\_path;?>/scripts.php">scripts</a><br>*

*<a href = "<?=$site\_path;?>/contact.php">contact</a><br>*

*</td>*

*<td valign="top" width="75%">*

*Welcome to PHPRecipes, the starting place for PHP scripts, tutorials, and information about gourmet cooking!*

*</td>*

*</tr>*

*</table>*

*Приклад 8*

*<?*

*// Файл: index.php*

*// Призначення: домашня сторінка PHPRecipes*

*// Дата: 23 серпня 2000 г.*

*// Вивести заголовок*

*include ("header.tpl");*

*// Вивести основну частину*

*include ("index\_body.tpl");*

*// Вивести колонтитул*

*include ("footer.tpl");*

*?>*

### *Приклад 9*

*<html>*

*<head>*

*<title> PHPRecipes </title>*

*</head>*

*<body bgcolor="#7b8079" text="#ffffff" link="#e7d387" alink="#e7d387" vlink="#e7f0e4">*

*<table width = "95%" cellpadding="0" cellspacing="0" border="1">*

*<tr>*

*<td valign = "top">*

*PHP Recipes*

*</td>*

*<td valign = "top" align="right">*

*August 23, 03:17 pm*

*</td>*

*</tr>*

*</table><table width="95%" cellspacing="0" cellpadding="0" border="1">*

*<tr>*

*<td valign="top" width="25%">*

*<a href = "http://localhost/phprecipes/tutorials.php">tutorials</a><br>*

*<a href = "http://localhost/phprecipes/articles.php">articles</a><br>*

*<a href = "http://localhost/phprecipes/scripts.php">scripts</a><br>*

*<a href = "http://localhost/phprecipes/contact.php">contact</a><br>*

*</td>*

*<td valign="top" width="75%">*

*Welcome to PHPRecipes, the starting place for PHP scripts, tutorials, and gourmet cooking tips and*

*recipes!*

*</td>*

*</tr>*

*</table><table width="95%" cellspacing="0" cellpadding="0" border="1">*

*<tr><td valign="top" align="middle">*

*Copyright &copy; 2000 PHPRecipes. All rights reserved.<br>*

*<a href = "mailto:wj@hotmail.com">contact</a> | <a href =*

*"http://localhost/phprecipes/privacy.php">your privacy</a>*

*</td></tr>*

*</table>*

*</body>*

*</html>*

*Приклад 10*

*<?*

*@mysql\_connect("localhost", "web", "ffttss")*

*or die("Could not connect to MySQL server!");*

*@mysql\_select\_db("company")*

*or die("Could not select products database!");*

*// Вибрати всі записи з таблиці products*

*$query = "SELECT \* FROM products";*

*$result = mysql\_query($query);*

*$x = 0;*

*print "<table>\n";*

*print "<tr>\n<th>Product ID</th><th>Product Name</th><th>Product Price</th>\n</tr>\n";*

*while ($x < mysql\_numrows($result)) :*

*$id = mysql\_result($result, $x, 'prod\_id');*

*$name = mysql\_result($result, $x, 'prod\_name');*

*$price = mysql\_result($result, $x, 'prod\_price');*

*print "<tr>\n";*

*print "<td>$id</td>\n<td>$name</td>\n<td>$price</td>\n";*

*print "</tr>\n";*

*$x++;*

*endwhile;*

*print "</table>";*

*mysql\_close();*

*?>*

*Приклад 11*

*<table>*

*<tr>*

*<th>Product ID</th><th>Product Name</th><th>Product Price</th>*

*</tr>*

*<tr>*

*<td>*

*1000pr*

*</td>*

*<td>*

*apples*

*</td>*

*<td>*

*1.23*

*</td>*

*</tr>*

*<tr>*

*<td>*

*1001pr*

*</td>*

*<td>*

*oranges*

*</td>*

*<td>*

*2.34*

*</td>*

*</tr>*

*<tr>*

*<td>*

*1002pr*

*</td>*

*<td>*

*Bananas*

*</td>*

*<td>*

*3.45*

*</td>*

*</tr>*

*<tr>*

*<td>*

*1003pr*

*</td>*

*<td>*

*pears*

*</td>*

*<td>*

*4.45*

*</td>*

*</tr>*

*</table>*

### *Приклад 12*

*<?*

*// Якщо змінна $bgcolor існує*

*if (isset($bgcolor)) :*

*setcookie("bgcolor", $bgcolor, time()+3600);*

*?>*

### *Приклад 13*

*<html>*

*<body bgcolor="<?=$bgcolor;?>">*

*<?*

*// Значення $bgcolor не задано, відобразити форму*

*else :*

*?>*

*<body bgcolor="white">*

*<form action="<? print $PHP\_SELF; ?>" method="post">*

*What's your favorite background color?*

*<select name="bgcolor">*

*<option value="red">red*

*<option value="blue">blue*

*<option value="green">green*

*<option value="black">black*

*</select>*

*<input type="submit" value="Set background color">*

*</form>*

*<?*

*endif;*

*?>*

*</body>*

*</html>*

*<?*

### *Приклад 14*

*if (! isset($userid)) :*

*$id = 15;*

*setcookie ("userid", $id, time()+3600);*

*print "A cookie containing your userID has been set on your machine.*

*Please refresh the page to retrieve your user information";*

*else:*

*@mysql\_connect("localhost", "web", "4tf9zzzf") or die("Could not connect to MySQL server!");*

*@mysql\_select\_db("user") or die("Could not select user database!");*

*// Об’явити запит*

*$query = "SELECT \* FROM users13 WHERE user\_id = '$userid'";*

*// Виконати запит*

*$result = mysql\_query($query);*

*// Якщо співпадання буде знайдено, вивести дані користувача.*

*if (mysql\_num\_rows($result) == 1) :*

*$row = mysql\_fetch\_array($result);*

*print "Hi ".$row["fname"].",<br>";*

*print "Your email address is ".$row["email"];*

*else:*

*print "Invalid User ID!";*

*endif;*

*mysql\_close();*

*endif;*

*?>*

# Лабораторна робота №6

## Вивчення мови програмування РНР, роботи з функціями файлової системи.

### Завдання:

*Написати програму, яка виконує такі дії:*

*1. Відобразити дерево інформації .*

*2. Приклад відображення:*

*3. Кожному файлу поставити у відповідність його розмір в байтах.*

*4.Програма повинна передбачити можливість руху по дереву інформації, перехід на рівень вище і перехід в кореневу папку.*

# Лабораторна робота №7

## Вивчення MySQL

***Мета роботи:****Вивчення основних функцій MySQL для написання PHP програм роботи з базами даних.*

*1. Послідовність роботи з таблицями в пакеті MySQL (з'єднання з сервером → вибір БД → обробка запитів → закриття з'єднання з сервером)*

*2. Основні функції MySQL (mysql\_connect, mysql\_select\_db, mysql\_query, mysql\_affected\_rows, mysql\_num\_rows, mysql\_result, mysql\_fetch\_row, mysql\_close)*

### Завдання

*Вивчення мови програмування РНР, робота з MySQL.*

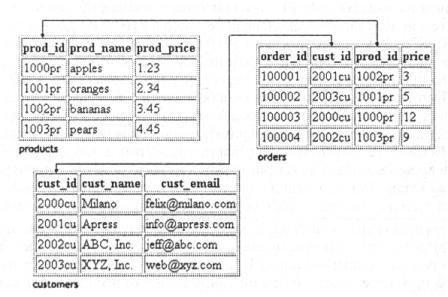
***Вихідні дані:***

*hostname="localhost";    
username="root";   
password="";*

*Ім’я бази даних = students*

*таблиці: products, orders, customers*

***Структура таблиці:***

**

*Необхідно побудувати SQL запит до БД і відобразити результат запиту в браузері в табличному вигляді.*

*Також вивести на екран кількість рядків у вихідний таблиці.*

### Теоретичні відомості:

*Розглянемо приклади скриптів для додавання, видалення, редагування та пошуку даних в базі даних з наступними полями:*

*- Ім'я - user\_name*

*- Телефон - phone*

*- Фотографія - img (ім'я файлу, який знаходиться в каталозі foto /)*

***Скрипт обробки загрузочних даних.***

*‹?php   
/\* Визначаємо значення для змінних\*/   
$hostname="localhost";   
$username="Ваш\_логін";   
$password="Ваш\_пароль";   
/\* ім’я бази даних \*/   
$dbName="Ім’я\_вашої\_бази\_даних";   
/\* Ім’я таблиці  \*/   
$usertable="Ім’я\_таблиці";   
/\* Створити з’єднання з MySql\*/   
MYSQL\_CONNECT($hostname,$username,$password) OR DIE("Не можу підключитися");   
/\* Вибір БД \*/   
MYSQL\_SELECT\_DB($dbName) or die("Не можу вибрати БД");   
/\* Введення інформації в БД \*/   
$query="INSERT INTO $usertable VALUES ('$user\_name','$phone','$img')";   
$result=MYSQL\_QUERY($query);   
/\* Закрити з’єднання \*/   
MYSQL\_CLOSE();   
print "Запис збережений в БД! ‹br›";   
?›****Скрипт обробки пошукового запросу.***

*‹?php   
$hostname="localhost";   
$username="Ваш\_логін";   
$password="Ваш\_пароль";   
$dbName="Ім’я\_вашої\_бази\_даних";   
$usertable="Ім’я\_таблиці";   
/\* Створити з’єднання з MySql\*/   
mysql\_connect($hostname,$username,$password) or die("No connect");   
mysql\_select\_db("$dbName") or die("No select DB");   
/\* Вибрати людей, імена яких починаються на $usr\_name \*/   
$query="select \* from $usertable where user\_name like '$user\_name%'";   
$result=mysql\_query($query);   
/\* Скільки знайшлось таких співробітників? \*/   
$number=mysql\_numrows($result);   
/\* Роздруківка вибраних співробітників \*/   
$i=0;   
if ($number == 0)   
{   
print "Немає даних о $user\_name в БД ‹br›";   
}   
elseif ($number › 0)   
{   
while ($i ‹ $number)   
{   
/\* Присвоєння змінним значень полів user\_name, phone, img для i-го  
рядка \*/   
$user\_name=mysql\_result($result,$i,"user\_name");   
$phone=mysql\_result($result,$i,"phone");   
$img=mysql\_result($result,$i,"img");   
/\* Вивід знайдених значень \*/   
print "‹br›‹img src='foto/$img' height=130›‹br›";   
print "‹u› Ім’я:‹/u› ‹br›";   
print "$user\_name ‹br›‹br›";   
print "‹u› Телефон: ‹/u›‹br›";   
print "$phone ‹br›‹br›";   
$i++;   
print "‹hr›";   
}   
}   
?›****Скрипт для видалення даних.***

*‹?php   
$hostname="localhost";   
$username="Ваш\_логін";   
$password="Ваш\_пароль";   
$dbName="Ім’я\_вашої\_бази\_даних";   
$usertable="Ім’я\_таблиці";   
mysql\_connect($hostname,$username,$password) or die("No connect");   
mysql\_select\_db("$dbName") or die("No select DB");   
/\* Видалення запису для поля user\_name \*/   
$query="delete from $usertable where user\_name='$user\_name'";   
$result=mysql\_query($query);   
print "Запис видалений ‹br›";   
?›****Скрипт редагування даних.***

*‹?php   
$hostname="localhost";   
$username="Ваш\_логін";   
$password="Ваш\_пароль";   
$dbName="Ім’я\_вашої\_бази\_даних";   
$usertable="Ім’я\_таблиці";   
mysql\_connect($hostname,$username,$password) or die("No connect");   
mysql\_select\_db("$dbName") or die("No select DB");   
/\* Вибір запису по полю user\_name \*/   
$query="select \* from $usertable where user\_name='$user\_name'";   
$result=mysql\_query($query);   
/\* Визначення наявності вибраного запису \*/   
$number=mysql\_numrows($result);  
if ($number==0)   
{   
print "Немає такого запису!‹br›";   
}   
else   
{   
/\* Якщо запис вибрана з бази даних, вивести всі її поля в форму для   
редагування і видалити її \*/   
$phone=mysql\_result($result,0,"phone");   
$img=mysql\_result($result,0,"img");   
print "‹img src='foto/$img' height=130›‹br›";   
/\* Видалення запису \*/   
$query="delete from $usertable where user\_name='$user\_name'";   
$result=mysql\_query($query);   
/\* Створення форми для редагування \*/   
print "‹form method=post action='form5.php'›";   
print "‹br›‹b›Введіть Ваше ім’я:‹/b›";   
print "‹br›‹input name='user\_name' value='$user\_name' size=30›";   
print "‹br›‹br›‹b›Введіть номер вашого телефону:‹/b›";   
print "‹br›‹input name='phone' value='$phone' size=10›";   
print "‹br›‹br›‹b›Введіть ім’я фото лат. буквами:‹/b›";   
print "‹br›‹input name='img' value='$img' size=20›";   
print "‹br›‹br›‹input type='submit' value='Зареєструвати'›";   
print "‹/form›";   
}   
?›****Скрипт для загрузки відредагованих даних.*** *‹?php   
hostname="localhost";   
$username="Ваш\_логін";   
$password="Ваш\_пароль";   
$dbName="ім’я\_вашої\_бази\_даних";   
$usertable="Ім’я\_таблиці";   
/\* Створити з’єднання \*/   
MYSQL\_CONNECT($hostname,$username,$password) OR DIE("Не можу підключитися");   
/\* Вибір БД \*/   
MYSQL\_SELECT\_DB($dbName) or die("Не можу вибрати БД");   
/\* Введення інформації в БД \*/   
$query="INSERT INTO $usertable VALUES ('$user\_name','$phone',''$img')";   
$result=MYSQL\_QUERY($query);   
/\* Закрити з’єднання \*/   
MYSQL\_CLOSE();   
print "Запис введена в БД! ‹br›";   
?›*