

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

Кафедра обчислювальної техніки та систем управління

**ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ
БАЗОВИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ**

МЕТОДИЧНІ ВКАЗІВКИ

**до виконання лабораторної та самостійних робіт
з дисципліни**

«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

Частина 5

Харків – 2018

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри обчислювальної техніки та систем управління 25 березня 2017 р., протокол № 8.

Методичні вказівки розроблено відповідно до робочої програми спеціальності «Електроенергетика, електротехніка та електромеханіка» з дисципліни «Обчислювальна техніка та програмування».

Метою методичних вказівок є набування необхідних знань з тематичних блоків: «Робота з електронною поштою»; «Створення Web-сторінок»; змістового модуля «Робота з Internet» та навичок використання отриманих теоретичних засад при розробленні програмних проєктів.

Рекомендується для студентів механіко-енергетичного факультету всіх форм навчання.

Укладачі:

доц. С. С. Бантюков,
старш. викл. І. Г. Бізюк,
доц. В. С. Меркулов,
асист. О. В. Чаленко

Рецензент

проф. В. І. Мойсеєнко

ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ БАЗОВИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної та самостійних робіт
з дисципліни
«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

Частина 5

Відповідальний за випуск Бізюк І. Г.

Редактор Еткало О. О.

Підписано до друку 13.09.17 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 1,5. Тираж 50. Замовлення №

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.

ЗМІСТ

Самостійна робота 1. Вивчення роботи електронної пошти.....	4
Самостійна робота 2. Макет і форматування тексту Web-сторінок.....	7
Самостійна робота 3. Вставка графічних зображень у Web-сторінку.....	13
Самостійна робота 4. Посилання у Web-сторінках.....	17
Самостійна робота 5. Таблиці у Web-сторінках.....	20
Лабораторна робота. Елементи управління у Web-сторінках...	29
Список літератури.....	34

Самостійна робота 1

ВИВЧЕННЯ РОБОТИ ЕЛЕКТРОННОЇ ПОШТИ

Мета: набути практичних навичок з використання програми Outlook Express для роботи з електронною поштою: створення, відправлення та отримання повідомлень.

Обладнання: персональний комп'ютер, програма Outlook Express.

Основні терміни і поняття: адресація комп'ютерів у мережі, основні поля поштового повідомлення (адреса, одержувач, тема), прикріплення (attachment) файлів, читання і відправлення повідомлень.

Завдання: відпрацювати в умовах локальної мережі порядок створення та відправлення поштових повідомлень одному або декільком користувачам.

1 Відправте повідомлення одному користувачеві.

2 Отримавши повідомлення, дайте відповідь на нього.

3 Надішліть копію вашого поштового повідомлення декільком користувачам одночасно.

4 Надішліть копію вашого поштового повідомлення декільком користувачам одночасно з використанням «прихованої копії».

5 Відправте повідомлення з прикріпленим файлом.

6 Додайте в поштове повідомлення довільний рисунок.

Зміст звіту: тема роботи, мета роботи, ескізи підготовлених та відправлених повідомлень (копії екрана).

Порядок виконання роботи

1 Відправте повідомлення одному користувачеві

Завантажити клієнтську програму Outlook Express 5.0 для роботи з електронною поштою в режимі «Читання пошти». Ознайомтеся зі структурою Головного вікна програми Outlook Express 5.0 (рисунок 1).

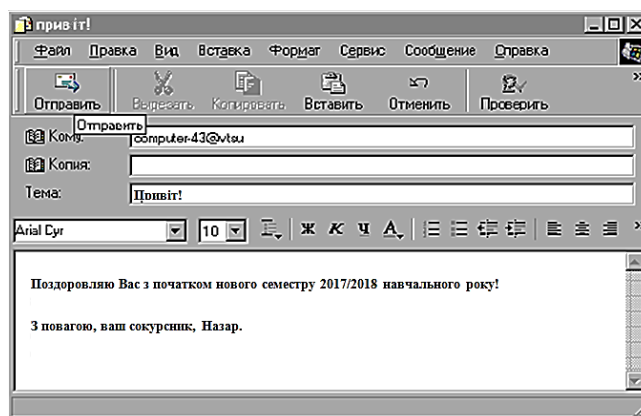


Рисунок 1 – Вікно програми Outlook Express 5.0

Знайдіть поля для введення поштової адреси одержувача, тему, текст повідомлення.

Створіть довільне повідомлення одного з користувачів, що перебувають у мережі. Для цього натисніть на панелі інструментів кнопку **Создать сообщение**. У полі **Кому** введіть адресу електронної пошти одержувача. Щоб вибрати одержувачів з адресної книги, натисніть на значок з книгою, що розміщений поруч з полями **Кому**, **Копия** та **СК** у вікні нового повідомлення, та виберіть потрібну адресу. У полі **Тема** введіть тему повідомлення. Введіть текст повідомлення та натисніть кнопку **Отправить** на панелі інструментів створюваного повідомлення.

У наявній локальній мережі постова адреса складається із власного імені комп'ютера (**computer-...**), номер якого вказано на системному блоці, і постійної частини, доменного імені сервера (**@vtsu**).

Примітка – Повідомлення можна надіслати самому собі, вказавши адресу свого комп'ютера.

2 Отримавши повідомлення, дайте відповідь на нього

Перевірте отриману пошту, відкривши папку **Входящие** (меню **Вид, Список папок**). Виберіть одне з отриманих повідомлень, подвійним натисканням відкрийте його для прочитання. Уважно вивчивши отриманий текст, натисніть кнопку **Ответить**. Перевірте ім'я адресата і тему. Зауважте, що частину вашого повідомлення вже написано автоматично. За необхідності видаліть зайві рядки автоматично створеної частини повідомлення, додайте свою думку і відправте адресату.

Для зручності створіть адресну книгу, внесіть до неї кількох користувачів і використовуйте її для адресації повідомлень.

3 Надішліть копію вашого поштового повідомлення декільком користувачам одночасно.

Відправлення повідомлення одночасно декільком користувачам виконується так само, як і відправлення повідомлення одному користувачеві. Адреси другого, третього і т. д. користувачів указуються в полі **Копія** через крапку з комою. Перевірте, в отриманому повідомленні повинно бути вказано, кому ще було відправлено повідомлення.

4 Надішліть копію вашого поштового повідомлення декільком користувачам одночасно із використанням «прихованої копії».

Поле **СК (Скрытая копия)** може бути відключено – увімкніть його в меню **Вид**. Перевірте, в отриманому повідомленні повинно бути вказано, кому ще було надіслано повідомлення, але без перерахування адрес із **СК**.

5 Відправте повідомлення з прикріпленим файлом

6 Додайте в поштове повідомлення довільний рисунок

Підготуйте за допомогою текстового редактора звіт, який містить опис виконаних дій, екранні форми і т. п.

Контрольні питання

- 1 Назвіть можливості і переваги електронної пошти.
- 2 Що являє собою лист електронної пошти?
- 3 Що таке «вкладення» до листа?
- 4 У яких форматах можуть відправлятися поштові повідомлення? За допомогою якої команди це встановлюється?
- 5 Як відповісти на отримане повідомлення?
- 6 Як відправити повідомлення декільком адресатам?
- 7 Як відправити копію або приховану копію?
- 8 Як зберегти отриманий разом із повідомленням файл?

9 Як створити файл з рисунком і відправити його по електронній пошті?

10 Як занести в адресну книгу інформацію про адресата?

11 Чи можна відправити повідомлення без теми?

Самостійна робота 2

МАКЕТ І ФОРМАТУВАННЯ ТЕКСТУ WEB-СТОРИНОК

Мета: вивчити основні принципи роботи мовою HTML. Засвоїти спільну роботу із додатками Блокнот, Internet Explorer, прийоми розмітки тексту і створення простих Web-сторінок.

Обладнання: персональний комп'ютер, програма Блокнот, браузер Internet Explorer.

Основні терміни і поняття: структура документа HTML, теги (парні і непарні, що відкривають і закривають), теги форматування тексту та їх атрибути (накреслення, вирівнювання, заголовки, шрифт, розмір, колір).

Завдання. створити власну Web-сторінку, використовуючи різні текстові елементи. Як зміст сторінки можете використовувати текст-презентацію вашої фірми; вас, як генерального представника цієї фірми, або інший довільний текст.

Сторінка повинна включати заголовок вікна Web-сторінки, заголовок тексту, елементи, набрані курсивом, напівжирним шрифтом, з різним вирівнюванням абзацу, різним колірним оформленням.

Наприклад: заголовок – ім'я або назва фірми (тег <h1>), підзаголовок – посада або позиція на ринку. Потім три-чотири абзаци, що використовують різні шрифти, колір, розмір. Для термінів або головних за змістом слів можна змінити накреслення. Наприкінці сторінки поставити підпис – вирівнювання по правому краю.

Зміст звіту: тема роботи, мета роботи, текст (набраний в Блокноте), ескіз створеної сторінки (копія екрана).

Порядок виконання роботи

1 Сумісна робота в Блокноті і в Internet Explorer

Створіть нову папку для розміщення html-документів. Ім'я папки повинно включати прізвище студента і групу, наприклад *Іванов 8- ЕТс*.

Запустіть **Блокнот**. У вікні додатка **Блокнот** наберіть заготовку сторінки як показано на рисунку 2.

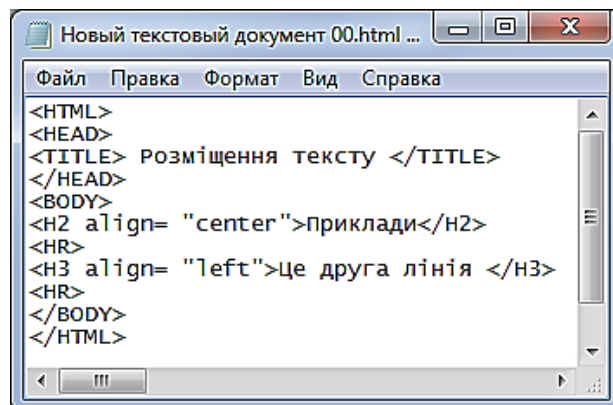


Рисунок 2 – Текст Web-сторінки в Блокноті

Збережіть текст у створеній папці під довільним іменем із розширенням **html**. Запустіть **Internet Explorer**. З допомогою команди **Открыть** меню **Файл** відкрийте створений вами файл. Ви повинні отримати зображення як на рисунку 3.

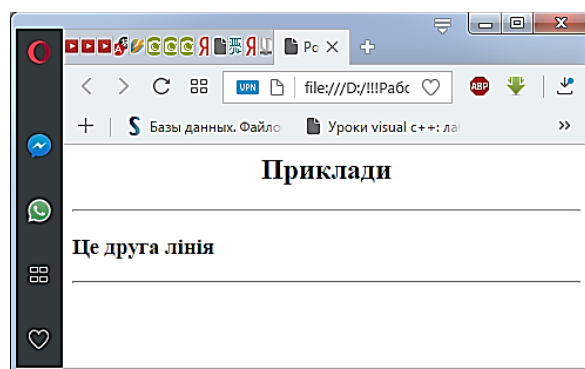



Рисунок 3 – Вид відповідної Web-сторінки в Internet Explorer

Змініть в **Блокноті**

рядок `<TITLE>Розміщення тексту</TITLE>`

на рядок `<TITLE>Сторінка Іванова</TITLE>`.

Знов збережіть текстовий файл. Перейдіть в **Internet Explorer**. Виберіть команду **Обновить** в меню **Вид** (кнопка  на панелі інструментів). Перевірте результат.

Діючи таким чином, ви зможете вносити зміни і доповнення в текст Web-сторінки і тут же спостерігати відповідні зміни в **Internet Explorer**.

2 Основні елементи сторінки HTML. Теги, що визначають структуру документа

Мова розмітки HTML (HyperText Markup Language) дає змогу користувачам створювати власні Web-сторінки. Кожна сторінка може містити складові частини – **елементи** (текст, заголовки, абзаци, рисунки, таблиці тощо). Основною одиницею розмітки сторінки в html є **тег** – певний текст, взятий у кутові дужки, наприклад **
. Теги можуть бути парними, тоді спочатку повинен бути записаний тег, що відкриває, наприклад **<i>, а наприкінці – тег, що закриває, з косою рисою-слешем, наприклад **</ i>**. Парні теги іноді називають ще **контейнерами**. Контейнери можуть містити в собі один або декілька елементів. Втім, деякі теги можуть бути непарними, тоді їм не потрібен тег, що закриває.

Загальна схема побудови контейнера у форматі html:

< ім'я _ тегу список _ атрибутів >

вміст контейнера – елементи

< / ім'я _ тегу >

Базова структура документа html повинна розташовуватися всередині елемента **<HTML> </HTML>**. Цим ми показуємо браузеру, що у файлі зберігається дійсно html-код. Далі в ці теги повинні бути вкладені парні піделементи **<head> </head>** (заголовок вікна) і **<body> </body>** (основне поле вікна Web-сторінки).

Основні елементи, які можуть бути вкладені в **<head>**, – це **<title>** – текст заголовка вікна, **<meta>** – службова інформація, **<base>** – інформація про базові ресурси посилань, **<script>** – програма для створення якого-небудь ефекту. У цій роботі використовується тільки тег **<title>**.

3 Основні теги форматування тексту

Найбільш часто застосовуються парні теги зміни накреслення тексту: `<i>` курсив `</i>`, `` **напівжирний текст** ``, `<i>` **напівжирний курсив** `</i>` і `<u>` підкреслений текст `</u>`.

Для початку нового абзацу застосовується тег `<p>`. Оскільки початок нового абзацу передбачає закінчення попереднього, тег `</p>`, що закриває, можна опустити. Для тегу `<p>` можна визначити параметр **align** – вирівнювання, яке може бути **left** (зліва), **right** (праворуч) та **center** (по центру). Починаючи новий абзац, браузер зазвичай робить відступ від попереднього абзацу в один рядок.

Примусово почати новий рядок можна з допомогою непарного тегу `
`. Текст при цьому друкується впритул, без відступу рядків, як це робилося на початку нового абзацу.

Крім звичайного тексту, у документі часто використовуються **заголовки**, що об'єднують за змістом розділи документа. Для цього призначені теги `<h1>` `</h1>` (найбільші за розміром), `<h2>` `</h2>` і так далі. Природно, кожен заголовок буде починатися з нового рядка. Вирівнювання в рядку заголовка визначається параметром **align**.

Приклади:

`<H2 align=center>` Заголовок другого рівня, вирівнювання по центру `</H2>`

`<p align=right>` Новий абзац з вирівнюванням по правому краю

4 Теги шрифту `` ``

Тепер перейдемо до універсального тегу форматування – `` ``. Він завжди повинен містити хоча б один з атрибутів **size**, **face**, **color**.

Розмір шрифту в html зазвичай задається не в пунктах або пікселях, а в умовних одиницях (не плутати з умов. од.). За замовчуванням розмір тексту встановлений у 3 умовні одиниці. При бажанні його можна збільшити в абсолютному виразі:

``

або ж у відносному виразі

``

Атрибут **face** визначає гарнітуру (назву) шрифту. Наприклад:

``

У цьому випадку буде використано шрифт **arial**.

У цьому випадку буде використано шрифт **Tahoma**, а якщо такого шрифту немає на цьому комп'ютері – то шрифт **Verdana**.

5 Основні теги управління кольором

Зміна кольору, звичайно, задається атрибутом **color**. Цей атрибут можна задавати кодовим словом – **red, green, blue, white, black, yellow, brawn, pink, gray, violet, orange, khaki** (хакі – пісочний).

Також можна кодувати колір набором шістнадцяткових чисел від 00 (відсутність кольору) до FF (максимальна насиченість кольору). Перші дві цифри коду позначають червоний колір, наступні дві цифри – зелений колір і останні дві цифри – синій колір. Наприклад:

**** (зелений)

або

**** (синій колір тексту на пісочному фоні)

Змінити колір можна також усередині тегу **<body>** з використанням атрибутів **text** (колір тексту сторінки), **bgcolor** (колір фону сторінки), **link** (колір гіперпосилання), **alink** (колір активного гіперпосилання), **vlink** (колір відвіданого гіперпосилання). Наприклад:

<body text = blue bgcolor=#F0e68c >

Результат – текст синього кольору на пісочному фоні.

6*¹ ASCII-графіка

Ще один корисний тег форматування – **<pre> </pre>**. Він призначений для друку тексту «як є», тобто так, як введено в Блокноті. Нарисуйте символами Блокнота будиночок (машинку, собачку і т. д.) і додайте її в текст. Наприклад,

¹ підвищеної складності

```

<pre>
  o
  o
 / \ ||
 / \ |
 /   \
-----
|     |
|     |
-----
</pre>

```

Контрольні питання

- 1 Що таке гіпертекстовий документ?
- 2 Назвіть основні синтаксичні конструкції мови HTML.
- 3 Дайте визначення понять «тег», «елемент», «контейнер».
- 4 Назвіть основні теги.
- 5 Запишіть приклад використання тегів, що визначають структуру документа.
- 6 У якій частині екрана виводиться напис, який поміщений у тег <Title> Документ </Title>?
- 7 У якому з тегів (що відкриває або що закриває) можуть задаватися атрибути?
- 8 Назвіть засоби, які дають змогу змінювати розміри і шрифти написів, що виводяться на сторінці.
- 9 Назвіть засоби, що дають змогу управляти кольором тексту і фону.

Самостійна робота 3

ВСТАВКА ГРАФІЧНИХ ЗОБРАЖЕНЬ У WEB-СТОРІНКУ

Мета: вивчити основні принципи додавання графічних зображень до Web-сторінки. Освоїти основні теги й атрибути, що визначають положення і розмір зображення.

Обладнання: персональний комп'ютер, програма Блокнот, Internet Explorer.

Основні терміни і поняття: теги форматування тексту та їх атрибути (накреслення, вирівнювання, заголовки, шрифт, розмір, колір).

Завдання: створити власну Web-сторінку, використовуючи різні текстові елементи. Як вміст сторінки можете використовувати текст-презентацію вашої фірми; вас, як генерального представника цієї фірми, або інший довільний текст.

Сторінка повинна містити графічний елемент (зображення) невеликого розміру (наприклад логотип фірми) з альтернативним текстом. Крім цього, як фон сторінки повинно бути використано інше зображення.

Зміст звіту: тема роботи, мета роботи, текст, набраний у Блокноте, ескіз створеної сторінки (копія екрана).

Порядок виконання роботи

1 Додавання графічних елементів

Попередні зауваження. *Сторінка (сайт) не повинна являти собою яскравий зразок графічної вишуканості, який важить сотні кілобайтів, бо така сторінка буде завантажуватися дуже довго.*

Для того, щоб вставити зображення в документ, використовується тег **img** (image – зображення) з атрибутом **src**, у якому вказується ім'я файлу рисунка.

``

Замість `my.jpg` можна підставити ім'я будь-якої картинки (`me.gif`, `main.jpg`). Основне – все, що розташовано між лапками, – посилання (шлях до картинки). Наш приклад говорить про те, що картинка лежить у тому ж каталозі (директорії, папці), у якій лежить і наш документ. Якщо картинка лежить в піддиректорії, то посилання на неї буде мати такий вигляд:

``

Якщо картинка лежить на рівень вище, а документ міститься у піддиректорії, то посилання на неї починатиметься з двох крапок:

```

```

Якщо картинка лежить на іншому сайті, то й шлях прописується повністю:

```

```

Для зручності кладіть картинку в ту ж директорію, що і документ, тоді буде менше плутанини.

2 Основні атрибути тега IMG

Найбільш уживаним є параметр **align**. Він визначає розташування (обтікання) тексту відносно зображення.

```

```

Параметр **left** означає, що картинка буде притиснута до лівого краю екрана, а текст обтікає її праворуч. Щоб зробити навпаки (картинка праворуч, текст зліва), треба прописати **right**:

```

```

Допустиме також обтікання зображення текстом знизу (за замовчуванням) – **bottom**, посередині – **middle** або зверху – **top**:

```

```

```

```

```

```

Крім параметра **align**, існують ще кілька параметрів.

Параметр **alt** – короткий опис картинки. Якщо навести курсор миші на рисунок і потримати кілька секунд, з'явиться фраза коментарю. У прикладі це фраза «моя фотографія». Якщо з якої-небудь причини зображення не з'явилося (наприклад відключене виведення графіки в браузері Internet), цей коментар виведе корисну інформацію.

```

```

Параметр **vspace** задає відстань між текстом і рисунком (по вертикалі). Відстань задається в пікселях. Pixel – мінімальна одиниця зображення, точка. Наприклад, розділення екрану 800x600 – 800 на 600 точок. У прикладі відстань становить 10 пікселів.

```

```

Параметр **hspace** задає відстань між текстом і рисунком по горизонталі. Відстань задається в пікселях. У прикладі вона дорівнює 30 пікселям (точкам).

```

```

Параметр **width** і **height** – ширина чи висота зображення в пікселях. Якщо параметр не ставити спеціально, то за замовчуванням ширина (висота) буде дорівнювати реальній ширині (висоті) картинки. З допомогою цих параметрів можна масштабувати зображення.

```

```

```

```

При відключенні графіки в Internet зображення до заданих параметрів **width** і **height** буде відображатися у вигляді порожнього прямокутника відповідних розмірів, без даних параметрів – невеликим значком (хрестиком).

Параметр **border** – рамка синього кольору навколо зображення (у пікселях).

Усі параметри можуть вживатися одночасно один з одним. Наприклад:

```

```

Зображення буде притиснуте до лівого краю екрана, текст обтікає його праворуч, відстань до тексту по горизонталі – 30 пікселів, по вертикалі – 5 пікселів (щоб красиво все виглядало), і якщо ви наведете курсор на картинку, то з'явиться напис – «моя фотографія».

При створенні власної Web-сторінки спробуйте поваріювати параметрами в атрибутах зображення, задавати різне значення в пікселях тощо – це допоможе вам краще засвоїти матеріал.

Для того, щоб розташувати саму картинку в центрі екрана (праворуч, ліворуч), можна використовувати теги параграфів (<p></p>) з атрибутом align.

3 Додавання зображення до фону Web-сторінки

Як уже зазначалося, атрибут **background**, що відкриває тег **BODY**, відповідає за стан фону. Як правило, зазначаються і фонове зображення **background**, і колір фону **bgcolor** – на той випадок, якщо фонове зображення з будь-яких причин не завантажиться (рисунок 4).

```
<body text=" #336699" bgcolor=" #000000" background=" фоновая_картинка.jpg" >
```


Наприклад

```
<HTML>
<HEAD>
<TITLE> Моя перша сторінка </TITLE>
</HEAD>

<BODY text=" #000000" bgcolor=khaki background=" lvbkgnd.jpg"
>
<H1 align=" center" > Про себе </H1>
<H2 align=" left" > Життєвий шлях </H2>

<p> Вчилася та працюю теж у <b>Харкові</b>. Чому і люблю це місто.

<p> Люблю і поважаю рідний Політехнічний інститут (нині – університет), якому присвятила чимало днів і ночей (перед іспитами). Добре ставлюся до людей, з якими працюю, співпрацювала і просто весело проводила час.
<HR>
<H2 align=" left" > Хобби і захоплення </H2>
<p> Основне захоплення – звичайно, робота. Не тільки тому, що це дає засоби до існування, але і тому, що дійсно це цікаво.

<p> З позаробочих захоплень – комп'ютер, як інструмент, як світ і як розвага. Книги, особливо хороших авторів з добре написаним словом. Музика, знову-таки, хороша і в хорошому виконанні.
<p> <i> Хороше в моєму розумінні означає – професійно зроблене, без халтури, з повагою до мене, як до споживача (слухача, читача).</i>
<HR>
<H2 align=" right" >І.Бізюк</H2>

</BODY>
</HTML>
```

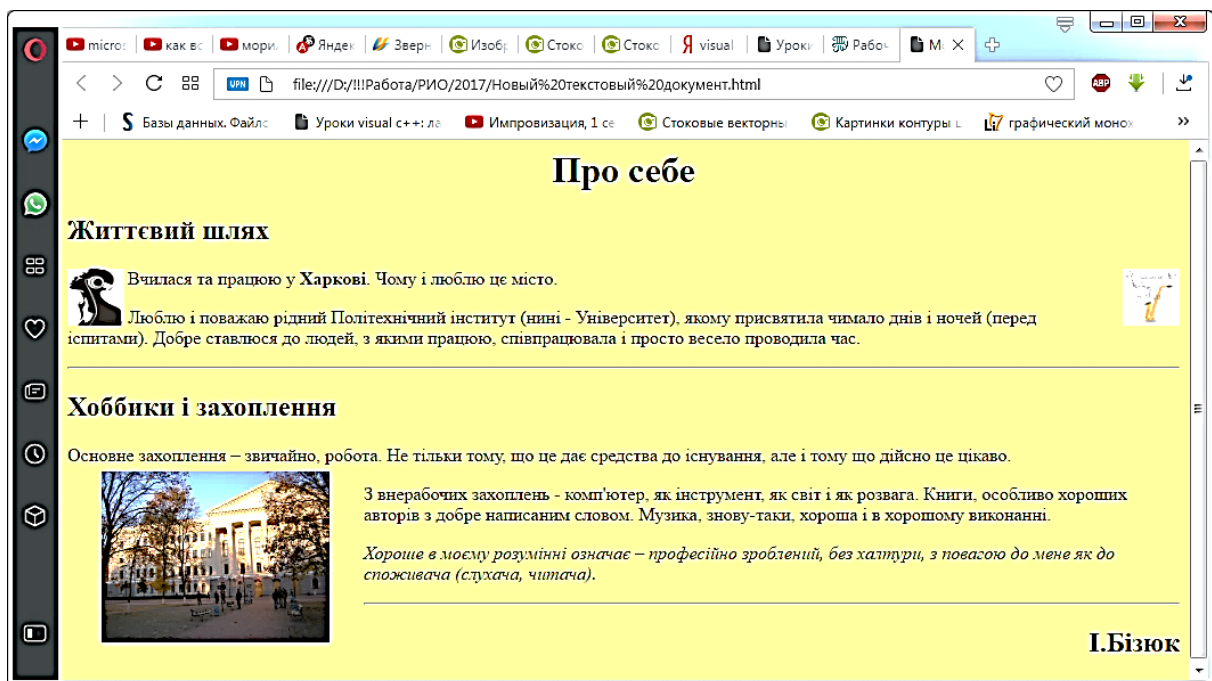


Рисунок 4 – Web-сторінка (Приклад)

Контрольні питання

- 1 Яким чином проводиться додавання зображень на Web-сторінку?
- 2 Для чого використовується атрибут alt?
- 3 Як змінити обтікання зображення текстом?
- 4 Яким чином задаються розміри зображення?
- 5 Яким чином можна змінити тло сторінки?

Самостійна робота 4 ПОСИЛАННЯ У WEB-СТОРИНКАХ

Мета: вивчити основні принципи створення посилань у Web-сторінках. Освоїти основні теги й атрибути, що визначають порядок переходу до інших сторінок.

Обладнання: персональний комп'ютер, програма Блокнот, Internet Explorer.

Основні терміни і поняття: теги посилань, порядок створення текстових посилань і посилань-рисуноків.

Завдання: обміняйтеся створеними вами сторінками з вашими товаришами. Використовуйте програму Провідник і

копіювання через диск або через локальну мережу. Зберіть у своїй папці 5-6 сторінок.

Створіть головну Web-сторінку, що використовує посилання-рисунок на вашу особисту сторінку (створену вами на минулому занятті), а також текстові посилання на сторінки ваших друзів.

Зміст звіту: тема роботи, мета роботи, текст, набраний у Блокноте, ескіз створеної сторінки (копія екрана).

Порядок виконання роботи

Поговоримо про посилання. Наша сторінка може складатися з декількох документів. Один з них – головний (index.html або main.html). Він відкривається першим і повинен обов'язково лежати на вашому сайті в Інтернеті.

Інші документи ви можете називати як завгодно (photos.html, about_me.html, my_pets.html, friends.html, gh516hgd.html). Вони всі можуть лежати в одній директорії (папці), а можуть у різних.

Посиланням на ці та інші документи (частини нашої сторінки) може бути текст (фраза, слово), а може бути і картинка. Для початку ми розглянемо тільки текстове посилання.

Створимо новий документ (у нашому прикладі prf.html у тій же директорії (папці), де міститься наш головний документ index.html. Зміст документа вигадайте самі, у вас для цього вже достатньо знань. Нехай friend.html – сторінка вашого друга. Зробимо посилання на friend.html:

```
<a href=" friend.html">Перейти на сторінку мого друга</a>
```

Тег <a> робить посиланням вкладений у нього рисунок або фразу (текст). Принципи прописування шляху тут такі ж, як у випадку з рисунками:

(1) - мои фотографии

(2) - мои фотографии

(3) - мои фотографии

У випадку (1) документ лежить у тій же директорії (папці), що й документ, у якому ми посилаємося на prf.html, у випадку (2) документ лежить у піддиректорії /photos, у випадку (3) ми

посилаємося на сайт <http://www.homepage.ru>, де лежить потрібний нам документ.

Як бачите, все просто. Але є один фактор, який нам потрібно врахувати: колір посилання потрібно прописувати, а інакше він буде просто жахливим. Тому згадаємо про тег BODY, що відкриває:

```
<body text="#336699" bgcolor="#000000" link="#339999"
      alink="#339999" vlink="#339999">
```

Розберемося що до чого: **link** – колір посилання, **alink** – колір активного посилання (натиснутого), **vlink** – колір уже відвіданого посилання. У нашому прикладі кольори однакові, але вони можуть бути різними. Колір посилання може бути таким же, як і колір тексту документа (посилання ж виділяється рисою). Отже, пропишемо кольори посилання і саме посилання в нашому документі.

```
<html>
<head>
<title> Мій перший крок </title>
</head>
<body text=" #336699" bgcolor=" #000000" link=" #339999"
alink=
" #339999" vlink=" #339999" >
<H3> Привіт, це моя перша сторінка.</H3>
<br>
<font color=" #CC0000" > Ласкаво просимо!</font>
<p align=" justify" >
Я дружу з Василем Петровим. Якщо ви хочете дізнатися про
нього більше – <a href=" prf.html" > прочитайте його сторінку
</a>,
</body>
</html> (прим.)
```

Як уже згадувалося, посиланням може бути і картинка. Принцип посилання той же, що і у випадку з текстом, тільки між тегами вставляється не текст, а рисунок.

```
<a href="фотки.html">  </a>
```

Таким чином, ми можемо зробити посиланням на документ з фотографіями (фотки.html) рисунок my_photo.gif. Таку позицію можна також вставити в текст вашої головної сторінки.

Контрольні питання

- 1 Яким чином проводиться створення посилань на Web-сторінку?
- 2 Як можна зробити посиланням зображення (рисунок)?
- 3 Для чого має сенс задавати колір посилання?
- 4 Чому для посилання можна визначити три кольори?
- 5 Як можна повернутися на початкову сторінку?

Самостійна робота 5 ТАБЛИЦІ У WEB-СТОРІНКАХ

Мета: вивчити основні принципи створення і використання таблиць у Web-сторінках. Освоїти основні теги й атрибути, що визначають порядок формування й об'єднання комірок таблиць.

Обладнання: персональний комп'ютер, програма Блокнот, Internet Explorer.

Основні терміни і поняття: теги посилань, порядок створення текстових посилань і посилань-рисуноків.

Завдання

1 Для закріплення навичок роботи з таблицями створіть просту таблицю розміром 2 рядки x 3 стовпці, розфарбувавши комірки різними кольорами, і помістіть у кожен комірку одне-два довільних слова.

2 Створіть складну таблицю, що являє собою план-схему вашої квартири (вашого офісу). Для зручності попередньо накресліть її на папері. У кожній комірці, що являє собою певну кімнату (кабінет), зробіть відповідний напис.

Зміст звіту: тема роботи, мета роботи, текст, набраний у Блокноте, ескіз створеної сторінки (копія екрана).

Порядок виконання роботи

Очевидно, ви вже працювали в Word і стикалися з таблицями. Якщо так, то ви повинні знати, що таблиці потрібні для форматування тексту, а часом вони незамінні.

Таблиця задається тегом:

```
<table></table>
```

На жаль, це далеко не все: таблиця складається з рядків і стовпців, комірок, тому нам треба ще вказати і їх.

`<tr></tr>` – рядок таблиці

`<td></td>` – стовпець (комірка) таблиці

На рисунку 5 подана таблиця з двох рядків і трьох стовпців (комірок). Межі таблиці не визначено, тому ви їх не бачите. Ця структура задається послідовністю тегів.

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

Рисунок 5 – Структура таблиці (Приклад)

Спочатку задаємо рядок. У нашому прикладі їх два:

`<table>` Створення таблиці в цілому

`<tr></tr>` Створення 1-го рядка

`<tr></tr>` Створення 2-го рядка

`</table>` Закінчення таблиці

Тепер у кожному рядку задамо по три стовпці (комірки):

```
<table>
```

```
<tr>
```

```
<td></td>
```

 Створення 1-ї комірки в 1-му рядку

```
<td></td>
```

 Створення 2-ї комірки в 1-му рядку

```
<td></td>
```

 Створення 3-ї комірки в 1-му рядку

```
</tr>
```

```
<tr>
```

```
<td></td>
```

 Створення 1-ї комірки в 2-му рядку

```
<td></td>
```

 Створення 2-ї комірки в 2-му рядку

```
<td></td> Створення 3-ї комірки в 2-му рядку
</tr>
</table>
```

Додаються теги-команди, виділені жирним шрифтом.

Порада: для початку нарисуйте таблицю на аркуші паперу, щоб усе наочно бачити.

Отже, тепер нам треба заповнити словами отриманий каркас:

```
<table>
<tr>
<td>1x1</td>
<td>1x2</td>
<td>1x3</td>
</tr>
<tr>
<td>2x1</td>
<td>2x2</td>
<td>2x3</td>
</tr>
</table>
```

Перша цифра в написах – номер рядка, а друга – номер стовпця (1x2 – перший рядок, другий стовпець і т. д.). Це знову ж таки для наочності. Якщо подивитися те, що вже у нас з вами вийшло, то це буде виглядати так, як показано на рисунку 6.

1x1 1x2 1x3
2x1 2x2 2x3

Рисунок 6

На жаль, фону (голубенького або жовтенького) ще не видно. Фон задається параметром `bgcolor="цвет_фона"`. Фон можна задати для таблиці в цілому, для рядка, для стовпця (у межах одного рядка). У нашому випадку ми задаємо фон для кожного стовпця (рисунок 7).

```
<table>
<tr>
```

```
<td bgcolor=" #FFCC33" >1x1</td>
<td bgcolor=" #336699" >1x2</td>
<td bgcolor=" #FFCC33" >1x3</td>
</tr>
```

```
<tr>
<td bgcolor=" #336699" >2x1</td>
<td bgcolor=" #FFCC33" >2x2</td>
<td bgcolor=" #336699" >2x3</td>
</tr>
</table>
```

Спробуйте задати фон для таблиці і для рядка (для засвоєння матеріалу).

Тепер згадаємо про параметри height і width – ви можете їх задати для всієї таблиці, для одного рядка, для комірки (стовпця). Згадайте, що висота і ширина можуть задаватися як у пікселях, так і відсотках. У нашому випадку ми задаємо ширину і висоту в пікселях для стовпців (комірок).

```
<table>
<tr>
<td height=" 35" width=" 50" bgcolor=" #FFCC33" > 1x1
</td>
<td width=" 50" bgcolor=" #336699" > 1x2 </td>
<td width=" 50" bgcolor=" #FFCC33" > 1x3 </td>
</tr>
<tr>
<td height=" 35" width=" 50" bgcolor=" #336699" > 2x1
</td>
<td width=" 50" bgcolor=" #FFCC33" > 2x2 </td>
<td width=" 50" bgcolor=" #336699" > 2x3 </td>
</tr>
</table>
```

Якщо в рядку задати для якої-небудь комірки (стовпця) висоту більшу, ніж для інших, то, незважаючи на різницю у висотах, усі комірки (стовпці) вашого рядка стануть по висоті дорівнювати найбільшому. Те ж саме з рядками: якщо ви поставите

для якогось рядка найбільшу довжину, то всі інші рядки вирівнюються по цьому найбільшому рядку. Пам'ятайте, рядок – це не комірка, тому в прикладі задана ширина для кожної комірки (стовпця) окремо.

Ви можете задати висоту і ширину всієї таблиці, тоді всі комірки (стовпці) і рядки поділять даний їм простір порівну, якщо не вказувати їм цей простір персонально, у відсотках від загальної ширини (висоти) таблиці або пікселях.

Повернемося до нашого прикладу. Тепер нам залишилося лише вирівняти вміст усередині таблиці:

```
<table>
<tr>
<td height=" 35" width=" 50" bgcolor=" #FFCC33" > <center>
1x1 </center> </td>
<td width=" 50" bgcolor=" #336699" > <center> 1x2 </center>
</td>
<td width=" 50" bgcolor=" #FFCC33" > <center>1x3 </center>
</td>
</tr>
<tr>
<td height=" 35" width=" 50" bgcolor=" #336699" > <center>
2x1 </center> </td>
<td width=" 50" bgcolor=" #FFCC33" > <center> 2x2 </center>
</td>
<td width=" 50" bgcolor=" #336699" > <center> 2x3 </center>
</td>
</tr>
</table>
```

У кожній комірці (стовпці) можуть перебувати і картинки, і текст, і навіть таблиці (у цьому випадку вони мають назву вкладені таблиці). І теги, які ми застосовуємо для форматування тексту, ті ж самі.

Оскільки вміст кожної комірки як би обстановка окремої кімнати, то і теги для центрування тексту довелося прописати в нашому прикладі в кожній комірці.

Повернемося до нашого прикладу, і поговоримо про вертикальне вирівнювання вмісту таблиці, тобто про те, як можна

зробити так, щоб вміст комірки не тільки розташовувався рівно посередині комірки (за замовчуванням), а ще вгорі або внизу. Вертикальне вирівнювання задається таким атрибутом: `valign="middle"` (`top`, `bottom`) – вміст конкретної комірки перебуватиме всередині комірки (вгорі чи внизу):

```
<table>
<tr>
<td height=" 35" width=" 50" bgcolor=" #FFCC33" valign="
top" > <center>1x1</center> </td>
<td width=" 50" bgcolor=" #336699" > <center>1x2</center>
</td>
<td width=" 50" bgcolor=" #FFCC33" valign=" bottom" >
<center>1x3</center> </td>
</tr>
<tr>
<td height=" 35" width=" 50" bgcolor=" #336699" valign="
bottom" > <center>2x1</center> </td>
<td width=" 50" bgcolor=" #FFCC33" > <center>2x2</center>
</td>
<td width=" 50" bgcolor=" #336699" valign=" top" >
<center>2x3</center> </td>
</tr>
</table>
```

Параметр **valign** прописаний не для всіх комірок, а тільки для тих, де текст розташовується зверху або знизу на розсуд автора (рисунок 7).

1x1	1x2	1x3
2x1	2x2	2x3

Рисунок 7

Тепер розглянемо більш складний приклад. Поговоримо про параметри **colspan** і **rowspan**. **Colspan** визначає кількість стовпців, на які простягається комірка, а **rowspan** – кількість рядків (ці параметри можуть набувати значення від 2 і більше, тобто наша

комірка може розтягуватися на два і більше стовпці (рядки)). Іншими словами, ці параметри задають об'єднання комірок.

Нагадаємо, таблиця, у якій у кожному рядку одна й та ж кількість стовпців, а в кожному стовпці – одна й та ж кількість рядків, називається регулярною таблицею. В іншому випадку – відповідно нерегулярною.

На рисунку 8 зображено план стандартної 2-кімнатної квартири. Для зручності світлими лініями показано кордони комірок у **регулярній** таблиці, з якої за допомогою об'єднання ми зробимо **нерегулярну** таблицю.

Гостиная		Спальня	
Коридор			
Ванная	WC	Кухня	

Рисунок 8

Розглянемо для початку перший рядок (Вітальня і Спальня). Очевидно, знадобляться теги `<table>`, `<tr>`, 2 теги `<td>` для двох комірок з параметром об'єднання `colspan=3` (для другої комірки = 2). Крім того, задамо для всієї таблиці рамку в 3 пікселі `border="3"`.

```
<table border=" 3" >
```

```
<tr>
```

```
<td height=" 150" width=" 250" colspan=" 3" >
```

```
<center>Гостиная </center> </td>
```

```
<td width=" 200" colspan=" 2" > <center> Спальня</center>
```

```
</td>
```

```
</tr>
```

```
</table>
```

Параметр **width** визначає загальну довжину комірки, без урахування того, скільки саме внутрішніх комірок брало участь у її об'єднанні (рисунок 9).

Гостиная	Спальня
----------	---------

Рисунок 9

Тепер пропишемо другий рядок. Коридор поки що розпишемо на дві комірки, третю займе кухня (рисунок 10).

```
<tr>
<td height=" 50" width=" 100" > <center> Коридор </center>
</td>
<td height=" 50" width=" 200" colspan=" 2" > <center> Коридор
</center> </td>
<td width=" 150" > <center> Кухня </center> </td>
</tr>
```

Гостиная		Спальня
Коридор	Коридор	Кухня

Рисунок 10

Далі врахуємо, що перший з двох коридорів займає цілих два рядки, так само, як і кухня. Для вертикального об'єднання використовується параметр `rowspan`. Принцип дії той же. Заодно і допишемо третій рядок, у якому додадуться всього лише дві комірки (Ванна і Туалет). Коридор і Кухню ми вже описали у другому рядку (рисунок 11).

```
<tr>
<td height=" 50" width=" 100" rowspan=2> <center> Коридор
</center> </td>
<td height=" 50" width=" 200" colspan=" 2" > <center> Коридор
</center> </td>
<td width=" 150" rowspan=2> <center> Кухня </center> </td>
</tr>
```

```

<tr>
<td height=" 100" width=" 100" > <center> Ванная </center>
</td>
<td width=" 100" > <center> WC </center> </td>
</tr>

```

Гостинная		Спальня	
Коридор	Коридор		Кухня
	Ванная	WC	

Рисунок 11

Останню дію, яку хотілося б зробити, – це прибрати розділову рису між двома коридорами. Справді, адже там немає стіни. Для цього для комірок з коридорами задамо білий колір кордону (рисунок 12).

```

<tr>
<td height=" 50" width=" 100" rowspan=2 bordercolor=white>
<center> Коридор </center> </td>
<td height=" 50" width=" 200" colspan=" 2 bordercolor=white" >
<center> Коридор </center> </td>
<td width=" 150" rowspan=2> <center> Кухня </center> </td>
</tr>

```

Гостинная		Спальня	
Коридор	Коридор		Кухня
	Ванная	WC	

Рисунок 12

Контрольні питання

- 1 Яким чином створюються таблиці на Web-сторінці?
- 2 Як можна додати до таблиці новий рядок?

- 3 Як можна в рядку таблиці додати ще одну комірку (стовпець)?
- 4 Як можна змінити колір заливки в комірці?
- 5 Як можна змінити колір кордону в комірці?
- 6 Як налаштовується вирівнювання тексту в комірці?
- 7 Якими параметрами визначається об'єднання комірок?

Лабораторна робота

ЕЛЕМЕНТИ УПРАВЛІННЯ У WEB-СТОРІНКАХ

Мета: вивчити основні принципи створення і використання елементів управління (кнопок, радіокнопок і т. д.) у Web-сторінках. Освоїти основні теги й атрибути, що визначають порядок створення і використання елементів управління, а також порядок написання елементарних функцій у Web-скриптах.

Обладнання: персональний комп'ютер, програма Блокнот, Internet Explorer.

Основні терміни і поняття: елементи управління, порядок створення елементів управління, написання скриптів.

Завдання: до початку лабораторної роботи підготуйте невеликий тест, у якому після відповідей на запитання можна дати деякий підсумковий висновок. Створіть Web-сторінку, на якій реалізовано тест опитування користувачів і виведення оброблених результатів у вигляді повідомлень.

Зміст звіту: тема роботи, мета роботи, текст, набраний у Блокноте, ескіз створеної сторінки (копія екрана).

Порядок виконання роботи

Ви вже звикли до того, що операційну систему необхідно попереджати, з яким саме об'єктом ви збираєтеся працювати. Тому першим рядком після стандартних тегів <HTML>, <HEAD>, <TITLE> і <BODY> необхідно оголосити використання форми:

<FORM action=" MyTest.html" method=" post" >

де MyTest.html – ім'я створюваної вами сторінки – тесту.

Далі можна і потрібно коротко описати ваш тест. Можете набрати щось на зразок:

<p> Для кожного питання вибрати найбільш відповідний варіант відповіді, який якнайкраще

описує ваше світосприйняття.
 Потім натисніть кнопку " Підсумки?"
, яка розміщена внизу екрана.

Нагадаємо, що тег
 створює м'який перехід на новий рядок.

Надалі потрібно записати питання тесту і передбачувані відповіді на них. Питання записується як звичайний абзац, а варіант відповіді оснащено радіокнопкою, натискання якої буде вказувати вибір користувача.

<P>1. Любите гарні вечірки?

<INPUT onclick=incHappy(1) type=radio value="" " name=question1>Так

<INPUT onclick=incUnhappy(1) type=radio value="" " name=question1> Не так, щоб дуже

Зверніть увагу. За умовами тесту підраховується окремо кількість позитивних (**Happy**) і негативних (**Unhappy**) відповідей на запитання. Відповідно в момент onclick на радіокнопці використовується та або інша функція збільшення =**incHappy(1)**. Цифра 1 у дужках означає, що відповідна змінна (кількість) збільшується саме на 1.

У тестах іншого типу у вас може бути тільки одна змінна, однак залежно від обраної відповіді в дужках може бути різна кількість балів.

Для прикладу наведемо опис ще одного запитання.

<P>2. А слабо подзвонити другу в 3 години ночі? Побалакати...

<INPUT onclick=incHappy(1) type=radio value="" " name=question2> Так запросто

<INPUT onclick=incUnhappy(1) type=radio value="" " name=question2> Не дуже

Зверніть увагу. Усі кнопки першого запитання мають параметр **name=question1**. Усі кнопки другого запитання мають параметр **name=question2**. Стежте, щоб у вашій сторінці ця тенденція не порушувалася.

Між запитанням і відповідями краще використовувати тег
, а перед початком нового запитання – тег <P>. Тим самим створюється невеликий відступ перед кожним запитанням.

Отже, усі запитання введені. Тепер потрібно написати функції обробки результатів. Усі ці функції повинні бути розташовані всередині тегу <SCRIPT> </SCRIPT>.

У початковому тегу <SCRIPT> необхідно вказати, якою саме мовою буде написаний цей скрипт. У цій лабораторній роботі пропонується використовувати мову JAVASCRIPT, як найбільш поширену в сучасному Інтернеті.

Перший рядок має задавати початок скрипту і використовувану мову.

```
<SCRIPT language=JAVASCRIPT type=TEXT/JAVASCRIPT>
```

Потім необхідно описати всі використовувані змінні (var). У нашому прикладі таких буде дві:

```
var Happy = 0
```

```
var Unhappy = 0
```

Їх початкове значення буде дорівнювати нулю. І це правильно, так як користувач ще не відповів на жодне запитання.

Потім слід описати функції, які ми використовували в радіокнопках. Їх структура елементарна, ви можете просто перезаписати цей текст у вашу сторінку, не забувши змінити ім'я функції. Зверніть увагу на фігурні дужки – вони дуже важливі в мові JAVASCRIPT.

```
function incHappy(points) {  
    Happy+=points;  
}
```

```
function incUnhappy(points) {  
    Unhappy+=points;  
}
```

Ще одна функція compute (обчислити) повинна обробляти результати тесту. У цьому прикладі порівнюються змінні **Happy** та **Unhappy**. Відповідно можливі три варіанти результатів, які виводяться повідомленням **window.alert**.

```
function compute(form)      ' Коментарі  
{                          ' це початок функції  
if (Happy>Unhappy) {      ' якщо більше відповідей Happy  
window.alert ("Ви щаслива людина"); ' вивести напис " щасливий"  
}                          ' кінець варіанта більше відповідей Happy  
else {                    ' інакше – але можливі ще 2 варіанти  
if (Unhappy>Happy) {     ' якщо більше відповідей Unhappy
```



```

window.alert ("Вам не завжди щастить у цьому житті"); 'вивести
напис "не дуже"
}
else
{
window.alert ("Головне – тримай баланс!"); вивести напис " серединка"
}
}
}
}
' кінець варіанта більше відповідей Harry
' інакше – перехід до третього варіанта
' початок третього варіанта
' кінець третього варіанта
' кінець розгляду якщо (IF)
' кінець усієї функції Compute

```

Як бачите, необхідно точно витримати кількість дужок, що закривають. Крім того, для системи важливі великі і маленькі букви, не плутайте їх.

В іншому прикладі покажемо, як можна обробляти одну змінну. Наприклад, якщо величина Harry більше 10, то це дуже добре, якщо менше 3, то це недобре, а серединка від 4 до 9 – нормально. Тоді функцію compute можна записати таким чином:

```

function compute(form)
{
if (Harry>10) {
window.alert ("Ви щаслива людина "); ' вивести напис " щасливий"
}
else {
if (Harry <3) {
window.alert ("Вам не завжди щастить у цьому житті ");
}
}
}
else
{
window.alert ("Головне – тримай баланс!"); ' вивести напис "
серединка"
}
}
}
}
' Комментарии
' це початок функції
' якщо Harry більше 10
' кінець варіанта Harry більше 10
' інакше – але можливі ще два варіанти
' якщо Harry менше 3
' вивести напис " не дуже"
' кінець варіанта Harry менше 3
' інакше – перехід до третього варіанта
' початок третього варіанта
' кінець третього варіанта
' кінець розгляду якщо (IF)
' кінець усієї функції Compute

```

У кінці потрібно закрити опис скрипту:

```
</SCRIPT>
```

Ще раз наведемо весь текст скрипту без коментарів.

```
<SCRIPT language=JAVASCRIPT type=TEXT/JAVASCRIPT>
var Harry = 0
```

```

var Unhappy = 0

function incHappy(points) {
    Happy+=points;
}

function incUnhappy(points) {
    Unhappy+=points;
}

function compute(form)
{
if (Happy>Unhappy) {
window.alert ("Ви щаслива людина ");
}
else {
if (Unhappy>Happy) {
window.alert ("Вам не завжди щастить у цьому житті");
}
else
{
window.alert ("Головне - тримай баланс!");
}
}
}
}
</SCRIPT>

```

Але на цьому робота не закінчена. Необхідно в кінці сторінки додати ще дві кнопки. Одна з них повинна буде запускати обробки результатів, а друга – видаляти результати тестування, якщо користувач зробив помилку.

Для першої кнопки можна записати такий тег:

```

<INPUT onclick=compute(form) type=button value=
" Ну і хто Я?" name=comput> </BUTTON>

```

Зверніть увагу. По-перше, змінився тип усталеного об'єкта порівняно з радіокнопкою. Тут `type=button`, а для радіокнопки було `type=radio`. По-друге, з'явився напис на кнопці `value=" Ну і хто Я?"`, а в радіокнопці такого напису не було (`value=""`). Зате після кнопки не виводиться ніякого тексту, тег, що закриває, `</BUTTON>` йде відразу за тегом, що відкриває, `<INPUT>`.

Друга кнопка може бути описана таким тегом:

```
<INPUT type=reset value=" Давай ще разок" name=clear>  
</BUTTON>
```

Тут тип type=reset – стандартний тип кнопки, який очищає зміни на формі. Те ж можна сказати про стандартне ім'я name=clear. Не забудьте закрити форму, тіло сторінки, розділ HTML.

```
</FORM> </BODY> </HTML>
```

Запускайте. Якщо ви ніде не помилилися в символах, усе повинно спрацювати відмінно!

СПИСОК ЛІТЕРАТУРИ

1 Фрейн, Б. Responsive Web Design with HTML5 and CSS3 [Текст] / Б. Фрейн – 2-е изд. – СПб. : Питер Пресс, 2017. – 272 с.

2 Вуль, В. А. Электронные издания [Текст] : / В. А. Вуль. – СПб. : БХВ–Санкт-Петербург, 2003. – 560 с.

3 Гасов, В. М. Методы и средства подготовки электронных зданий [Текст] : / В. М. Гасов, А. М. Цыганенко. – М. : МГУП, 2001. – 735 с.

4 Роббинс, Д. HTML5 : Pocket Reference [Текст] / Д. Роббинс. – 5-е изд. – М. : Вильямс, 2015. – 192 с.

5 Кит, Д. HTML5 for web designers [Текст] / Д. Кит – СПб.: Манн, Иванов и Фербер, 2015. – 112 с.

6 Інформатика. Комп'ютерна техніка. Комп'ютерні технології [Текст] : посібник / за ред. О. І. Пушкаря. – К. : Видавничий центр «Академія», 2001. – 696 с.

7 Кенцл, Т. Форматы файлов Интернет [Текст] / Т. Кенцл. — СПб. : Питер, 1997 – 320 с.

8 Коркин, А. MS Internet Explorer 6.0. Самоучитель [Текст] / А. Коркин. – СПб. : БХВ-Санкт-Петербург, 2002. – 288 с.

9 Круг, С. Веб-дизайн [Текст] / С. Круг, Дж. Блэк. – М. : Символ-Плюс, 2001. – 195 с.

10 Карнеги, Д. Как выработать уверенность в себе и влиять на людей, выступая публично [Текст] / Д. Карнеги. – М. : Современный литератор, 2007. — 320 с.

11 Прохоренок, Н. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера [Текст] / Н. Прохоренок, В. Дронов. – 4 изд. – СПб. : БХВ-Петербург, 2015. – 766 с.

12 Нильсен, Я. Web-дизайн [Текст] /Я. Нильсен. – М. : Символ-Плюс, 2000. – 512 с.

13 Пауэл Т.А. Web-дизайн [Текст] /Т. А. Пауэл. — СПб. : БХВ-Санкт-Петербург, 2004. –1072 с.

14 Симонович, С. Новейший самоучитель по работе в Интернете [Текст] /С. Симонович, Г. Евсеев. – М. : Изд. «ДЕСС КОМ», 2000. – 528 с.

15 Смирнова, И. Е. Начала Web-дизайна [Текст] /И.Е. Смирнова. — СПб. : БХВ-Санкт-Петербург, 2003. – 256 с.