

Центральноукраїнський національний технічний університет

Факультет обліку та фінансів

Кафедра економічної теорії, маркетингу та економічної кібернетики

WEB-ПРОГРАМУВАННЯ

методичні вказівки до вивчення дисципліни
для студентів всіх форм навчання за спеціальностями
051 “Економіка” (спеціалізація “Економічна кібернетика”) та 075 “Маркетинг”
(освітній рівень “бакалавр”)

Затверджено
на засіданні кафедри ЕТМЕК
Протокол № 12 від 25.04.2018

Web-програмування: методичні вказівки до вивчення дисципліни для студентів всіх форм навчання за спеціальностями 051 “Економіка” (спеціалізація “Економічна кібернетика”) та 075 “Маркетинг” (освітній рівень “бакалавр”) / [укл. к.е.н., доц. І.В. Ніколаєв, к.е.н., доц. Дмитришин Б.В.]. – Кропивницький : ЦНТУ, 2018. – 32 с.

Укладачі: Ніколаєв І.В. – к.е.н., доцент кафедри ЕТМЕК,
Дмитришин Б.В. – к.е.н., доцент кафедри ЕТМЕК

Рецензент: Вишневська В.А. – к.е.н., доцент кафедри ЕТМЕК

ЗМІСТ

Вступ	4
1. Структура, тематичний план та програма навчальної дисципліни	5
2. Опис складу змістовних модулів навчальної дисципліни	5
Лекції	7
Лабораторні заняття	11
3. Орієнтовна тематика рефератів з навчальної дисципліни	22
4. Контрольні питання з навчальної дисципліни	23
5. Рекомендована література	25
Додатки	28

ВСТУП

Всесвітня павутина (англ. World Wide Web) – розподілена система, що надає доступ до пов'язаних між собою документів, розташованих на різних комп'ютерах, підключених до Інтернету. Для позначення Всесвітньої павутини також використовують слово веб (англ. web “павутина”) і аббревіатуру WWW.

Всесвітню павутину утворюють сотні мільйонів Web-серверів. Більшість ресурсів всесвітньої павутини засновані на технології гіпертексту. Гіпертекст – текст, представлений у вигляді асоціативно зв'язаних автономних блоків. Гіпертекст дозволяє включати в сторінки посилання на інші частини даного ж або іншого документа. Це дозволяє організувати зв'язок між різними сторінками і об'єднати їх в єдину систему.

Гіпертекстові документи, що розміщуються у Всесвітній павутині, називаються Web-сторінками. Кілька Web-сторінок, об'єднаних загальною темою, дизайном, а також зв'язаних між собою посиланнями, які зазвичай знаходяться на одному і тому ж Web-сервері, називаються Web-сайтом. Для створення Web-сторінок та Web-сайтів використовують мову гіпертекстової розмітки HTML (HyperText Markup Language) та багато інших мов програмування.

У системі підготовки бакалаврів за спеціальностями 051 “Економіка” (спеціалізація “Економічна кібернетика”) та 075 “Маркетинг” дисципліна “Web-програмування” є базовою у формуванні комплексу знань і навичок із використання новітніх інформаційних технологій.

Швидкий розвиток і широке застосування засобів обчислювальної техніки визначають вимоги до підготовки сучасного фахівця, що повинен за допомогою мови HTML вміти створювати Web-документи, інтерактивні статичні та динамічні Web-сторінки, Web-сайти підприємств та установ.

Отже, **метою вивчення курсу “Web-програмування”** у вищій школі є ознайомлення студентів з теоретичними основами створення Web-документів на основі стандартів HTML; ознайомлення з областю застосування різноманітних стандартів та засобів створення Web-контенту, а також набутті навичок практичного створення Web-контенту різними засобами.

Курс “Web-програмування” передбачає такі **навчальні завдання**:

- оволодіння основними термінами та поняттями мови гіпертекстової розмітки HTML, технологіями Web-програмування з акцентом на розробку Web-сайтів для їх подальшого розміщення в глобальній мережі Інтернет;

- набуття навичок та вмінь застосування знань зі створення Web-документів та Web-сайтів для їх використання у професійній сфері.

У результаті вивчення навчальної дисципліни студенти повинні:

ознайомитися з теоретичними основами Web-технологій та гіпертексту; засобами структурного та візуального оформлення Web-документів; принципами об'єднання окремих Web-сторінок у Web-сайти та налагодження взаємодії з користувачами;

знати протоколи передачі даних в глобальній мережі Інтернет; принципи організації серверів у глобальних мережах; вимоги до дизайну Web-сторінок і принципи юзабіліті; принципи роботи з палітрами кольору; основні теги, використовувані при створенні Web-документів; теоретичні засади створення Web-сайтів за допомогою різних Web-технологій; принципи роботи з HTML-редактором;

вміти створювати Web-сторінки за допомогою текстового та HTML-редакторів; розробляти та ефективно застосовувати графічне оформлення сайтів; розробляти, створювати та підтримувати роботу сайтів; працювати з системами управління контентом.

Запропоновані методичні вказівки до вивчення дисципліни “Web-програмування” містять: структуру, тематичний план та програму курсу, тематику лекцій і лабораторних робіт, включаючи завдання для самостійної роботи студентів, тематику рефератів, список рекомендованої літератури, контрольні питання, за допомогою яких у студентів буде можливість перевірити свої знання.

1. СТРУКТУРА, ТЕМАТИЧНИЙ ПЛАН ТА ПРОГРАМА

НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
Тема 1. Поняття Web-технологій	8	2	-	-	-	6	16	0,5	-	-	-	12
Тема 2. Мова гіпертекстової розмітки HTML	12	2	-	4	-	6	15,8	0,5	-	0,5	-	12
Тема 3. Основні теги управління стилем тексту	14	2	-	4	-	8	16	1	-	1	-	15
Тема 4. Навігація в гіпертекстових документах	12	2	-	4	-	6	16	0,5	-	0,5	-	15
Тема 5. Графіка в гіпертекстових документах	12	2	-	4	-	6	15,6	0,5	-	0,5	-	15
Тема 6. Таблиці в гіпертекстових документах	14	2	-	4	-	8	16	1	-	0,5	-	15
Тема 7. Каскадні таблиці стилів CSS (Cascading Style Sheets)	14	2	-	2	-	8	17	-	-	0,5	-	16
Тема 8. Фрейми	10	-	-	2	-	8	15,3	-	-	-	-	15
Тема 9. Реалізація клієнтської активності	12	-	-	4	-	8	16	0,5	-	0,5	-	15
Усього годин	90	14	-	28	-	48	120	4	-	4	-	112

2. ОПИС СКЛАДУ ЗМІСТОВНИХ МОДУЛІВ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Поняття Web-технологій

- 1.1. Інтернет та його сервіси.
- 1.2. Принципи взаємодії браузера та Web-сервера.
- 1.3. Гіпертекст як основа WWW.
- 1.4. Види Web-сторінок.
- 1.5. Типи Web-вузлів.
- 1.6. Планування Web-вузла.
- 1.7. Загальні правила проектування навігаційної схеми сайту.
- 1.8. Ролі в Web-дизайні.

Тема 2. Мова гіпертекстової розмітки HTML

- 2.1. Загальні відомості про HTML.
- 2.2. Історія та версії HTML.
- 2.3. Тегова модель.

- 2.4. Парні та одиночні теги, контейнери.
- 2.5. Елементи HTML.
- 2.6. Класифікація елементів HTML.
- 2.7. Вкладені елементи HTML.
- 2.8. Атрибути тегів.
- 2.9. Основні правила синтаксису HTML.
- 2.10. Загальна структура HTML-документа.
- 2.11. META-дані та пошук.
- 2.12. Створення шаблону Web-сторінки.

Тема 3. Основні теги управління стилем тексту

- 3.1. Оформлення тексту.
- 3.2. Елементи для форматування тексту.
- 3.3. Елементи для форматування великих блоків тексту.
- 3.4. Заголовки.
- 3.5. Створення абзацу.
- 3.6. Примусове перенесення тексту на новий рядок.
- 3.7. Марковані списки.
- 3.8. Нумеровані списки.
- 3.9. Списки визначень.
- 3.10. Управління шрифтом.
- 3.11. Завдання базового шрифту.
- 3.12. Прокрутка тексту.
- 3.13. Коментарі в Web-документах.

Тема 4. Навігація в гіпертекстових документах

- 4.1. Створення гіперпосилань.
- 4.2. Внутрішні посилання.
- 4.3. Посилання в межах одного документу.
- 4.4. Зовнішні посилання.
- 4.5. Посилання на інші об'єкти.

Тема 5. Графіка в гіпертекстових документах

- 5.1. Вставка графічних зображень у гіпертекстові документи.
- 5.2. Способи вирівнювання графічних зображень.
- 5.3. Встановлення висоти та ширини графічних зображень.
- 5.4. Встановлення рамки навколо графічних зображень.
- 5.5. Встановлення відступів між графічним зображенням та текстом.
- 5.6. Графічне зображення – посилання.
- 5.7. Завдання графічного зображення у якості фону гіпертекстового документу.

Тема 6. Таблиці в гіпертекстових документах

- 6.1. Створення таблиць.
- 6.2. Елементи таблиць.
- 6.3. Визначення структури таблиці.
- 6.4. Завдання заголовку таблиці.
- 6.5. Встановлення рамки таблиці.
- 6.6. Встановлення відстані між рамкою таблиці та текстом.
- 6.7. Встановлення висоти та ширини таблиці.
- 6.8. Вирівнювання елементів таблиці.
- 6.9. Встановлення фонових кольору або малюнка таблиці.
- 6.10. Групування рядків та стовпчиків таблиці.

6.11. Вкладені таблиці.

Тема 7. Каскадні таблиці стилів CSS (Cascading Style Sheets)

- 7.1. Поняття про каскадні аркуші стилів.
- 7.2. Основні принципи CSS.
- 7.3. Можливості CSS.
- 7.4. Визначення вбудованого стилю.
- 7.5. Формування аркуша стилів.
- 7.6. Види селекторів.
- 7.7. Зовнішні аркуші стилів.
- 7.8. Стильові властивості.

Тема 8. Фрейми

- 8.1. Визначення структури фреймового документа.
- 8.2. Створення фреймового документу.
- 8.3. Завдання рядків та стовпчиків.
- 8.4. Встановлення розмірів фреймів.
- 8.5. Встановлення полів фреймів.
- 8.6. Налаштування скролінгу.
- 8.7. Налаштування переключення фреймів по гіперпосиланнях.

Тема 9. Реалізація клієнтської активності

- 9.1. Поняття форми.
- 9.2. Методи пересилки інформації GET та POST.
- 9.3. Створення полів для вводу рядків тексту, пароля, імені локального файлу.
- 9.4. Створення елементів-прапорців.
- 9.5. Створення елементів-перемикачів.
- 9.6. Створення кнопки пересилки інформації.
- 9.7. Створення кнопки скидання.
- 9.8. Створення елементів-графічних зображень.
- 9.9. Створення прихованих елементів.
- 9.10. Створення випадаючих списків.

ЛЕКЦІЇ – 14 год.

Тема1. Поняття Web-технологій – 2 год.

- 1.1. Інтернет та його сервіси.
- 1.2. Принципи взаємодії браузера та Web-сервера.
- 1.3. Гіпертекст як основа WWW.
- 1.4. Види Web-сторінок.
- 1.5. Типи Web-вузлів.
- 1.6. Планування Web-вузла.
- 1.7. Загальні правила проектування навігаційної схеми сайту.
- 1.8. Ролі в Web-дизайні.

Література [3; 4; 6; 7; 8; 13; 15; 19; 20; 23; 35; 39; 44]

Завдання до самостійної роботи

1. Основи Web-дизайну та принципи створення Web-сайтів.
2. Хостінг та просування сайтів.

Питання до самоконтролю:

1. Назвіть мету та завдання дисципліни “Web-програмування”.
2. Розкрийте поняття всесвітньої мережі Інтернет.

3. Назвіть основні сервіси мережі Інтернет.
4. Що таке Web-браузер та які вони бувають?
5. Що таке Web-сервер та для чого він потрібен?
6. Розкрийте поняття гіпертексту.
7. Розкрийте сутність служби WWW (World Wide Web).
8. Які види Web-сторінок ви знаєте?
9. Охарактеризуйте основні типи Web-вузлів.
10. Назвіть та опишіть етапи планування Web-вузлів.
11. Які загальні правила проектування навігаційної схеми сайту ви знаєте?
12. Розкрийте ролі у Web-дизайні.

Тема 2. Мова гіпертекстової розмітки HTML – 2 год.

- 2.1. Загальні відомості про HTML.
- 2.2. Історія та версії HTML.
- 2.3. Тегова модель.
- 2.4. Парні та одиночні теги, контейнери.
- 2.5. Елементи HTML.
- 2.6. Класифікація елементів HTML.
- 2.7. Вкладені елементи HTML.
- 2.8. Атрибути тегів.
- 2.9. Основні правила синтаксису HTML.
- 2.10. Загальна структура HTML-документа.
- 2.11. META-дані та пошук.
- 2.12. Створення шаблону Web-сторінки.

Література [1; 2; 4; 6; 10; 11; 12; 14; 16; 17; 18; 21; 28; 33; 38; 39; 46]

Завдання до самостійної роботи

1. Стандарт розширеної мови розмітки XML (eXtensible Markup Language).

Питання до самоконтролю:

1. Дайте визначення HTML (HyperText Markup Language).
2. Що називають розміткою документу?
3. Надати основні відомості з історії розвитку мови HTML.
4. Охарактеризувати тегову модель мови HTML.
5. Розкрийте поняття парного, одиночного тега, контейнера.
6. Що називають елементами HTML?
7. Навести класифікацію елементів HTML.
8. Що таке вкладені елементи HTML?
9. Наведіть правила вкладення елементів HTML.
10. Що називають атрибутами тегів та які правила їх запису?
11. Наведіть основні правила синтаксису HTML.
12. Опишіть загальну структуру HTML-документа.
13. Що називають META-даними та яку роль для пошуку вони відіграють?

Тема 3. Основні теги управління стилем тексту – 2 год.

- 3.1. Оформлення тексту.
- 3.2. Елементи для форматування тексту.
- 3.3. Елементи для форматування великих блоків тексту.
- 3.4. Заголовки.
- 3.5. Створення абзацу.
- 3.6. Примусове перенесення тексту на новий рядок.
- 3.7. Марковані списки.

- 3.8. Нумеровані списки.
- 3.9. Списки визначень.
- 3.10. Управління шрифтом.
- 3.11. Завдання базового шрифту.
- 3.12. Прокрутка тексту.
- 3.13. Коментарі в Web-документах.

Література [1; 2; 3; 5; 9; 10; 14; 16; 17; 18; 21]

Завдання до самостійної роботи

- 1. Методи кодування тексту на Web-вузлах.

Питання до самоконтролю:

- 1. Розкрийте базові правила оформлення тексту в HTML-документах.
- 2. Наведіть основні елементи для форматування тексту.
- 3. Наведіть елементи для форматування великих блоків тексту.
- 4. Розкрийте сутність заголовків та методи їх створення.
- 5. Опишіть процедуру створення абзацу в гіпертекстовому документі.
- 6. Яким чином виконується примусове перенесення тексту на новий рядок?
- 7. Яким чином можна заборонити перенесення тексту на новий рядок?
- 8. Що таке списки та які вони бувають?
- 9. Опишіть процедуру створення маркованого списку.
- 10. Опишіть процедуру створення нумерованого списку.
- 11. Опишіть процедуру створення списку визначень.
- 12. Яким чином здійснюється управління шрифтом в HTML-документах?
- 13. Що таке базовий шрифт та яким чином він задається?
- 14. Яким чином реалізується прокручування тексту?
- 15. Як задати коментар до фрагменту коду в HTML-документі?

Тема 4. Навігація в гіпертекстових документах – 2 год.

- 4.1. Створення гіперпосилань.
- 4.2. Внутрішні посилання.
- 4.3. Посилання в межах одного документу.
- 4.4. Зовнішні посилання.
- 4.5. Посилання на інші об'єкти.

Література [2; 3; 4; 5; 6; 8; 10; 11; 13; 16; 17; 19; 20; 35]

Завдання до самостійної роботи

- 1. Види навігації на Web-сайтах та засоби їх реалізації.

Питання до самоконтролю:

- 1. Розкрийте сутність гіперпосилання.
- 2. Які типи гіперпосилань ви знаєте?
- 3. Яким чином створюється гіперпосилання?
- 4. Опишіть створення внутрішніх гіперпосилань.
- 5. Опишіть особливості створення гіперпосилань в межах одного документу.
- 6. Опишіть створення зовнішніх гіперпосилань.
- 7. Наведіть приклади посилань на інші об'єкти.

Тема 5. Графіка в гіпертекстових документах – 2 год.

- 5.1. Вставка графічних зображень у гіпертекстові документи.
- 5.2. Способи вирівнювання графічних зображень.
- 5.3. Встановлення висоти та ширини графічних зображень.

- 5.4. Встановлення рамки навколо графічних зображень.
- 5.5. Встановлення відступів між графічним зображенням та текстом.
- 5.6. Графічне зображення – посилання.
- 5.7. Завдання графічного зображення у якості фону гіпертекстового документу.

Література [3; 10; 13; 18; 20; 21; 24; 29; 35; 39; 44; 45]

Завдання до самостійної роботи

1. Векторна і растрова графіка.
 2. Техніка створення Web-графіки Macromedia Flash.
- Питання до самоконтролю:*
1. Які графічні зображення можна розміщувати в гіпертекстових документах?
 2. Опишіть процедуру вставки графічних зображень у гіпертекстові документи.
 3. Які способи вирівнювання графічних зображень ви знаєте?
 4. Яким чином задається висота та ширина графічних зображень?
 5. Яким чином задається рамка навколо графічного зображення?
 6. Як задати альтернативний текст, що виводиться замість зображення?
 7. Як змінити відступи між графічним зображенням та текстом?
 8. Що потрібно для того, щоб перетворити графічне зображення у гіперпосилання?
 9. Як встановити фонове зображення у гіпертекстовому документі?

Тема 6. Таблиці в гіпертекстових документах – 2 год.

- 6.1. Створення таблиць.
- 6.2. Елементи таблиць.
- 6.3. Визначення структури таблиці.
- 6.4. Завдання заголовку таблиці.
- 6.5. Встановлення рамки таблиці.
- 6.6. Встановлення відстані між рамкою таблиці та текстом.
- 6.7. Встановлення висоти та ширини таблиці.
- 6.8. Вирівнювання елементів таблиці.
- 6.9. Встановлення фонових кольору або малюнка таблиці.
- 6.10. Групування рядків та стовпчиків таблиці.
- 6.11. Вкладені таблиці.

Література [2; 5; 14; 16; 17; 18; 20; 27; 31; 32; 36]

Завдання до самостійної роботи

1. Основи JavaScript.
 2. JScript як альтернатива мови JavaScript.
- Питання до самоконтролю:*
1. У яких випадках в гіпертекстових документах використовуються таблиці?
 2. З яких елементів складається таблиця?
 3. Як визначити структуру таблиці?
 4. Яким чином задається заголовок таблиці?
 5. За допомогою яких атрибутів встановлюється рамка таблиці?
 6. Як змінити відступи між рамкою таблиці та текстом?
 7. Які способи вирівнювання елементів таблиці ви знаєте?
 8. Яким чином задається висота та ширина таблиці?
 9. Як встановити фонові колір або малюнок таблиці?
 10. Для чого і яким чином виконується групування рядків та стовпчиків таблиці?
 11. Наведіть метод вкладки однієї таблиці в іншу.

Тема 7. Каскадні таблиці стилів CSS (Cascading Style Sheets) – 2 год.

- 7.1. Поняття про каскадні аркуші стилів.
- 7.2. Основні принципи CSS.
- 7.3. Можливості CSS.
- 7.4. Визначення вбудованого стилю.
- 7.5. Формування аркуша стилів.
- 7.6. Види селекторів.
- 7.7. Зовнішні аркуші стилів.
- 7.8. Стильові властивості.

Література [2; 9; 10; 11; 14; 16; 17; 18; 35; 46]

Завдання до самостійної роботи

1. Стильові властивості блоків.
2. Стильові властивості списків.

Питання до самоконтролю:

1. Розкрийте поняття каскадних аркушів стилів.
2. Що називають стильовими властивостями?
3. Які можливості та переваги надає використання CSS?
4. Дайте визначення вбудованого стилю.
5. Яким чином формується аркуш стилів?
6. Що таке визначення стилю, селектор?
7. Наведіть види селекторів.
8. Як формуються зовнішні аркуші стилів?
9. Наведіть відомі вам стильові властивості.

ЛАБОРАТОРНІ ЗАНЯТТЯ – 28 год.

Тема 1. Поняття Web-технологій – 2 год.

Лабораторна робота №1. Дослідження інтерфейсу та функціональних можливостей сучасних браузерів

Мета роботи: ознайомитись з призначенням та принципами роботи програм-браузерів. Дослідити їх функціональні можливості та елементи вікна.

Загальні положення.

Під браузером розуміють програму перегляду інформації в WWW. При цьому всі дані інформаційних ресурсів підготовлені в спеціальному гіпертекстовому форматі мовою HTML (HyperText Markup Language). Браузер є засобом перегляду, призначеним для виконання переходів і одержання доступу до інформаційних ресурсів служби WWW. На сьогоднішній день найбільш розповсюджені наступні Web-браузери: Internet Explorer, Mozilla Firefox, Google Chrome, Opera. Наступний розгляд функціональних елементів та можливостей буде проходити на прикладі браузера Internet Explorer.

Запуск та завершення роботи з програмою Internet Explorer

Залежно від конфігурації програмного забезпечення комп'ютера та операційного середовища існує кілька методів запуску програми Internet Explorer:

1. двічі клацнути піктограму Internet Explorer на Робочому столі Windows (при її наявності);
2. виконати команду Головного меню **Пуск – Програми – Internet Explorer**;
3. відкрити будь-який файл з розширенням **.htm** або **.html**, двічі клацнувши його мишею;
4. запустити програму Internet Explorer (файл iexplore.exe) на виконання командою **Пуск – Виконати – Обзор** (як правило, вона знаходиться за адресою C:\Program Files\Internet

Explorer\iexplore.exe);

5. двічі клацнути на Робочому столі ярлик, який представляє збережену раніше Web-сторінку. Після цього завантажиться програма Internet Explorer, у вікні якої відкриється відповідний документ.

Завершення роботи з програмою Internet Explorer здійснюється будь-яким способом, придатним для завершення роботи з програмним додатком Windows.

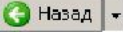
Рядок заголовку. Відображає інформацію про назву завантаженого ресурсу.


Адресний рядок. Містить URL (тобто Web-адресу) відображеного в браузері на даний момент документа.


Рядок стану браузера. Повідомляє про поточний стан браузера. Відбиває інформацію про різні стадії процесу завантаження документа: встановлення контакту з Web-сервером, передану частину документа в процентах та ін. По закінченню передачі документа в рядку міститься інформація про посилання, на який показує курсор мишки.


Смуги гортання. Це – звичні для Windows-додатків смуги гортання, потрібні для пересування поточним Web-документом.


Призначення кнопок панелі інструментів вікна Internet Explorer

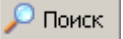
1.  **Back** (Назад) або комбінація клавіш **Alt+←**, або команда меню **Вид – Переход – Назад**. Кнопка **Back** дозволяє повернутися з поточного документа до попереднього. Ці два документи не обов'язково мають бути з'єднані гіпертекстовим посиланням. Читаючи документ А, можна завантажити документ Б, просто вводячи його адресу в адресний рядок. Тоді кнопка **Back** дозволить користувачеві повернутися назад в документ А. Якщо користувач за допомогою гіпертекстового посилання перейде до іншої частини того самого документа, кнопка **Back** поверне користувача на попереднє місце цього документа, а не до попереднього документа.

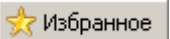
2.  **Forward** (Вперед) або комбінація клавіш **Alt+→**, або команда меню **Вид – Переход – Вперед**. Ця кнопка доступна тільки тоді, коли треба повернутися на один або кілька кроків назад кнопкою **Back**. Вона дозволяє відмінити повернення і знову завантажити наступний документ у ланцюжку, що його пройшов користувач. Такий самий результат дасть вибір такого самого, як і минулого разу, посилання на сторінці.

3.  **Остановить** (або клавіша **Esc**, або команда меню **Вид – Остановить**). Перериває завантаження документа.

4.  **Обновить** (або клавіша **F5**, або команда меню **Вид – Обновить**). Якщо одержання документа призупинено кнопкою **Остановить**, або документ завантажено не повністю, користувач може примусити браузер повторити спробу одержати документ, натиснувши кнопку **Обновить**. В цьому (і тільки в цьому) випадку браузер не звертає уваги на те, чи є вже в його кеші примірник цієї сторінки, а знову повторює всю послідовність дій (пошук сервера в Інтернеті, посилання запиту і одержання документа з мережі).


5.  **Домой** (Початкова сторінка). Цю кнопку треба натиснути, щоб повернутися до початкової сторінки. Її адресу користувач може визначити командою **Сервис – Свойства обозревателя**.


6.  **Поиск** (або комбінація клавіш **Ctrl+E**, або команда меню **Вид – Панели обозревателя – Поиск**). Клацнувши цю кнопку, користувач побачить на екрані Web-сторінку, з якої можна розпочати пошук інформації в WWW з використанням популярних пошукових систем.


7.  **Избранное** (або комбінація клавіш **Ctrl+I**, або команда меню **Вид – Панели обозревателя – Избранное**). Список **Избранное**, елементи якого інакше називаються закладками, є зручним способом впорядкування часто відвідуваних Web-сторінок, і створення на них посилань.

8.  **Медиа** (або команда меню **Вид – Панели обозревателя – Медиа**). Встановлення панелі **Медиа** в Internet Explorer спрощує відтворення музики, відео та

мультимедійних файлів.

9.  **Журнал** (або комбінація клавіш **Ctrl+N**, або команда меню **Вид – Панели обозревателя – Журнал**). У журналі автоматично фіксуються сторінки, відвідані раніше – сьогодні, вчора або кілька тижнів тому. Щоб проглянути яку-небудь сторінку, її слід вибрати з цього списку.

10.  **Почта**. Здійснюється виклик програмних засобів, які забезпечують підтримку роботи з електронною поштою.

11.  **Печать** (Вивести на принтер). Ця кнопка, як і команда **Файл – Печать**, дозволяє надрукувати поточну сторінку на принтері, приєднаному до комп'ютера користувача. На екрані з'являється традиційне вікно друку для Windows-додатків.

Порядок виконання роботи:

1. Завантажити будь-який Web-браузер, яким ви користуєтесь, всіма відомими вам способами. Описати ці способи у звіті.

2. Дослідити та описати у звіті структуру вікна Web-браузера та призначення всіх його елементів.

3. Дослідити та описати у звіті структуру меню вікна Web-браузера та призначення всіх команд меню. Особливу увагу приділити навігаційним засобам, наявним у меню.

4. Дослідити та описати у звіті структуру адресного рядка вікна Web-браузера та його призначення. Описати, як встановити або видалити з вікна адресний рядок; як за допомогою адресного рядка можна завантажувати Web-сторінки, які раніше вже завантажувались.

5. Дослідити та описати у звіті структуру рядка звичайних кнопок на панелі інструментів вікна Web-браузера. Описати, як встановити або видалити з вікна цей рядок. Особливу увагу приділити навігаційним засобам, наявним на панелі інструментів Web-браузера.

6. Дослідити та описати у звіті призначення журналу та технологію роботи з ним.

7. Дослідити та описати у звіті призначення команди меню **Избранное (Закладки)** та технологію роботи з його пунктами.

8. Дослідити та описати у звіті технологію збереження інформації, яка відображається у вікні Web-браузера, та окремих її елементів (тільки тексту, тільки графічного зображення, текстового фрагменту тощо). Проаналізувати, як саме зберігається текстова і графічна інформація з Web-сторінок.

9. Дослідити та описати у звіті можливості Web-браузера, які стосуються відображення інформації у його вікні (розмір шрифту, вибір способу кодування – ASCII, Windows-1251, KOI8, UTF-8 та ін.).

10. Відкрити початкову сторінку Web-вузла і переглянути її у вигляді html-коду (**Вид – просмотр HTML-кода**).

Питання до самоконтролю:

1. Дати визначення поняттю “Браузер”. Які браузери вам відомі?

2. Охарактеризувати призначення браузера який ви досліджували? Перелічити всі відомі вам способи завантаження досліджуваного браузера.

3. Описати елементи вікна браузера та їх призначення.

4. Охарактеризувати призначення команд меню вікна браузера.

5. Для чого призначається команда меню **Вид – Панели инструментов – Закрепить Панели инструментов** браузера Internet Explorer.

6. Як здійснюється додавання інформації в **Избранное (Закладки)**?

7. Як здійснюється впорядкування інформації, що зберігається в теці **Избранное**?

8. Як зупинити завантаження деякої сторінки?

9. Як поновити інформацію, що відображається на сторінці?

10. Як здійснюється налагодження властивостей браузера (встановлюється адреса початкової (домашньої) сторінки, налагоджується та очищується журнал, встановлюється режим видалення тимчасових файлів, змінюється оформлення вікна браузера тощо)?

11. Як надрукувати Web-сторінку?
12. Як зберегти Web-сторінку у вигляді .html-файлу?
13. Чим відрізняється інформація збережена у вигляді Web-сторінки (.html-файл) та Web-архіву (.mht-файл)?
14. Як переглянути зміст Web-вузла у вигляді html-коду?

Тема 2. Мова гіпертекстової розмітки HTML – 4 год.

Лабораторна робота №2. Планування Web-вузла та створення шаблону Web-сторінки

Мета роботи: практичне закріплення навичок студентів щодо планування Web-вузла та побудови шаблону Web-сторінки.

Загальні положення.

Планування Web-вузла та розробка його навігаційної схеми

Гіпертекстові сторінки (Web-сторінки) являють собою деяку зв'язуючу ланку між об'єктами різних типів та можуть бути статичними, інтерактивними або динамічними. Будь-який Web-вузол (сайт) складається з великої кількості гіпертекстових сторінок. Найбільш розповсюдженими типами Web-вузлів є персональна сторінка (блог), вузол організації, тематичний вузол, комерційний вузол.

Правильне планування Web-вузла значно спрощує всю подальшу роботу і гарантує успішне завершення проекту. Наприклад, якщо ми збираємось створити відносно складний Web-ресурс, то без планування дизайну сайт, швидше за все, не буде цілісним. Процес планування можна умовно розділити на шість етапів:

1. Визначення призначення вузла.

Мета – це той результат, який ми хочемо отримати з допомогою Web-вузла за певний період часу. **Завдання** – це дії, які слід зробити для досягнення поставлених цілей.

2. Опис аудиторії.

Цільова аудиторія – це група людей, на яку розраховане створення Web-вузла.

3. Планування вмісту.

В якості вмісту на вузлі може бути різний контент: текст, фотографії, зображення, анімація, відеоролики, звукові доріжки, дані що динамічно збираються.

4. Планування структури.

Ескіз Web-вузла допоможе розробити зручну для користувача систему навігації.

5. Планування Web-сторінок.

Для того щоб спланувати Web-сторінку, необхідно розуміти її функціональне навантаження. Так сторінка може бути домашньою або звичайною сторінкою Web-вузла.

6. Планування навігації.

Система навігації залежить від структурної схеми Web-вузла. Продумана навігаційна схема (меню) повинна надавати можливість оцінити, яка інформація знаходиться на сайті, і наскільки швидко до неї можна дістатися. Перебуваючи на будь-якій сторінці, відвідувач повинен представляти своє місце розташування, знати, що він може зробити на цій сторінці, і куди можна піти далі. При дотриманні деяких нескладних правил можна з легкістю створити меню, по якому складно буде заблукати:

- на головній сторінці Web-вузла повинна бути представлена карта сайту (меню) в графічному або текстовому вигляді;
- на всіх наступних сторінках повинні бути присутніми посилання на головну сторінку, на якій у свою чергу знаходяться посилання на ці ж сторінки;
- при використанні малюнків у якості посилань вони повинні бути однотипними на всіх сторінках і розташовуватися в одній і тій же позиції;
- малюнки, які використовуються в якості посилань (кнопки), повинні бути інтуїтивно помітні для відвідувача щодо інших елементів сторінки.

Для виконання навігації зазвичай застосовуються наступні типи посилань:

- внутрішні посилання, що зв'язують сторінки в межах одного сайту;
- контекстні посилання, всередині Web-сторінки між певними фрагментами інформації;
- зовнішні посилання, для зв'язку з документами поза межами даного сайту;
- посилання на інші об'єкти або електронну пошту.

Створення шаблону Web-сторінки

Для створення Web-сторінок зазвичай використовують мову гіпертекстової розмітки документів **HyperText Markup Language**, або скорочено – **HTML**.

Розмітка Web-документів виконується за допомогою тегів, які записуються з дотриманням певних правил. Тегова модель передбачає розбиття документа на окремі фрагменти, які укладаються в теги або починаються тегом.

Тег (tag – покажчик, мітка) – це код, що ідентифікує певний фрагмент Web-документа, наприклад, абзац, заголовок, посилання, таблицю і т.д. Теги укладаються у кутові дужки <>.

Загальна структура Web-документа наступна:

1. Рядок, представлений тегом **<!DOCTYPE>** призначений для зазначення типу поточного документа (містить інформацію для браузера про версію HTML, а також про те чи підтримуються застарілі елементи та атрибути або фрейми).

2. Рядок, що представлений тегом **<HTML>**, вказує браузеру, що далі слідує саме HTML-код (а не скажімо PHP).

3. Заголовна частина, що відкривається тегом **<HEAD>**. Це декларативна частина, в якій міститься назва документа (тег **<TITLE>**), теги для пошукових машин (**META**-теги), а також можуть розташовуватися службова інформація для серверів та опису сценаріїв.

4. Тіло документа, представлене елементом **BODY** або елементом **FRAMESET**.

Порядок виконання роботи:

1. Визначити призначення власного Web-вузла (сформулювати мету ресурсу, завдання, визначитися з його типом). За допомогою пошукових систем ознайомитись з типовими Web-вузлами у мережі Інтернет, що відповідають власній тематиці.

2. Провести опис цільової аудиторії Web-вузла. Для цього створіть профіль представника цільової аудиторії, вивчіть вимоги членів цільової аудиторії.

3. Розробити загальну концепцію та провести планування вмісту Web-вузла. Вирішіть якій контент (текст, графічні та динамічні елементи, таблиці, інші матеріали) найкращим чином підійдуть для досягнення поставленої мети.

4. За допомогою ескізів сплануйте структуру Web-вузла таким чином, щоб вона задовольняла цілям ресурсу. Продумати імена файлів Web-сторінок, шляхи для навігаційних посилань тощо.

5. Виконати планування Web-сторінок у залежності від їх функціонального навантаження (домашня, звичайна тощо). Продумайте яка саме інформація, у якому вигляді та у якому місці повинна бути присутня на Web-сторінках.

6. Розробити навігаційну схему Web-вузла та описати елементи навігації, що будуть використовуватися (текстові посилання, кнопки, мапи зображень, меню, система пошуку тощо).

7. Створити шаблон Web-сторінки. У шаблоні передбачити такі елементи: основні META-теги (кодування сторінки, ім'я автора, авторські права, ключові слова, короткий опис сторінки), назву Web-вузла, яка відображається у заголовку вікна браузера. Задати кольори сторінки, тексту, гіперпосилань, гарнітури шрифтів, розмір базового шрифту.

8. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Які види Web-сторінок ви знаєте?
2. Охарактеризуйте основні типи Web-вузлів.
3. Назвіть та опишіть етапи планування Web-вузлів.
4. Які організаційні питання стоять перед розробниками Web-вузлів?
5. Для чого призначений ваш Web-вузол та як буде визначатися його успішність?
6. Які загальні правила проектування навігаційної схеми сайту ви знаєте?
7. Які технічні проблеми необхідно вирішити для успішного “запуску” Web-вузла?

8. Де та на яких умовах можна розмістити готовий Web-вузол?
9. Опишіть загальну структуру Web-документа та призначення його тегів.
10. Що називають META-даними та яку роль для пошуку вони відіграють?

Тема 3. Основні теги управління стилем тексту – 4 год.

Лабораторна робота №3. Робота з текстовими елементами HTML

Мета роботи: навчитися створювати прості Web-сторінки з використанням мови HTML

Загальні положення.

Елементи для форматування тексту діляться на дві групи: логічні та фізичні.

Логічні елементи повідомляють браузеру про те, який тип інформації в них міститься, наприклад важливий текст або цитата. Браузер сам вирішує, як відобразити такий текст. Ці елементи розбивають документ на логічні частини і при цьому не зобов'язують браузер відображати текст, розташований всередині елемента, яким-небудь певним чином. До логічних елементів відносять наступні: **CITE**, **CODE**, **EM**, **Q**, **STRONG**, **VAR** тощо.

Фізичні елементи просто говорять браузеру, як повинен виглядати той чи інший блок тексту, не уточнюючи ніяк сенс і значимість його вмісту. Іншими словами, такий елемент змушує браузер намалювати букву червоною, напівжирною або курсивом, не уточнюючи, чому буква має виглядати саме так. До фізичних елементів відносять наступні: **B**, **I**, **TT**, **U**, **SUB**, **SUP** тощо.

Заголовки – важливий елемент сайту, вони допомагають систематизувати текст. Заголовки створюються за допомогою елементів **H1**, **H2**, **H3**, **H4**, **H5**, **H6**. За замовчуванням заголовок самого верхнього рівня виділяється самим великим шрифтом, чим нижче рівень заголовка, тим менше шрифт.

Абзаци, як відомо, ділять текст на логічні частини і на листі виділяються відступом від краю аркуша. У HTML абзаци відокремлюються один від одного відстанню в один рядок. Для організації абзаців в HTML передбачений елемент **P**, який передбачає наявність закриваючого тега.

Списки – важливий інструмент, вони застосовуються для організації і групування даних. Це може стати в нагоді при створенні карти сайту (тобто його змісту), опису складних структур та інших подібних об'єктів. В HTML можна виділити кілька типів списків: маркований, нумерований, список визначень.

Марковані списки – це списки, в яких пункти відзначаються за допомогою різних символів. Такі списки ще називають нумерованими, або неупорядкованими, тому що для елементів даного списку послідовність неважлива. Ці списки можна використовувати для простого перерахування об'єктів або їх властивостей. Для створення списків у HTML передбачений елемент **UL**, що вимагає наявності закриваючого тега, а всі пункти списку знаходяться всередині цього елемента.

Нумеровані списки застосовуються, коли порядок слідування пунктів списку має велике значення, наприклад при описі алгоритмів або інших покрокових дій. Особливістю списків цього типу є те, що всі їх елементи впорядковані. Для створення впорядкованих списків застосовується елемент **OL**, якому потрібна наявність закриваючого тега, а всі пункти списку знаходяться всередині цього елемента.

Список визначень – це особливий вид списку, який застосовується для форматування словників або коли необхідно пояснювати значення термінів. Особливістю списку визначень впливає з його функцій: елемент такого списку завжди складається з двох частин: слово, що визначається; опис або розшифровка терміну. Форматування проводиться таким чином, що опис терміна відображається з відступом від краю екрану і, можливо, з пропуском рядка від визначення. Для організації списків визначень служить елемент **DL** – всередині нього будуть знаходитися визначення та опис терміна.

Основні теги та їх атрибути призначені для форматування та управління стилем тексту документа наведені у додатку 1, а базові кольори представлені у додатку 2.

Порядок виконання роботи:

1. Підготувати декілька текстів які б відповідали меті та завданням власного Web-вузла (див. лабораторну роботу №2).

2. Засобами текстового редактора (блокнот) укласти підготовлені тексти в контейнер BODY шаблонів Web-сторінки розробленої у лабораторній роботі №2. Для цього скористатися тегами HTML призначеними для управління стилем тексту. У документах передбачити щонайменше: два рівні заголовків, один із варіантів списків, форматування абзацу, блоки тексту, оформлення тексту різними кольорами. За потреби, також можна використовувати розділові лінії, прокрутку тексту тощо.

3. Зберегти документи як файли із розширенням .txt, а потім змінити розширення на .html (або відразу зберегти документи як файли із розширенням .html, якщо це дозволяє текстовий редактор). При виборі імен документів використовуйте прості слова, що відображають їх зміст та відповідають структурі Web-вузла, наприклад: index.html, about.html, product.html, price.html, contacts.html тощо. В результаті буде отримано декілька Web-сторінок.

4. Провести тестування створених Web-сторінок у браузері. У разі необхідності відкоригувати їх, скориставшись переглядом змісту сторінки у вигляді html-коду.

5. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Розкрийте базові правила оформлення тексту в HTML-документах.
2. Наведіть основні елементи для форматування тексту.
3. Наведіть елементи для форматування великих блоків тексту.
4. Розкрийте сутність заголовків та методи їх створення.
5. Опишіть процедуру створення абзацу в гіпертекстовому документі.
6. Яким чином виконується примусове перенесення тексту на новий рядок?
7. Що таке списки та які вони бувають?
8. Яким чином здійснюється управління шрифтом в HTML-документах?
9. Яким чином реалізується прокручування тексту?
10. Як тестувати Web-сторінку?

Тема 4. Навігація в гіпертекстових документах – 2 год.

Лабораторна робота №4. Робота з гіперпосиланнями

Мета роботи: набути практичних навичок із встановлення зв'язків між документами та дослідити функціонування створених гіперпосилань.

Загальні положення.

Гіперпосилання пов'язують документи, розкидані по всьому Інтернету, в одну мережу. Сайт може перебувати на різних комп'ютерах, але для відвідувача він буде здаватися єдиним цілим, і все це завдяки посиланням. Для створення гіперпосилань в HTML служить елемент **A**, який вимагає наявності закриваючого тега. Усередині елемента розташовується текст, який буде виділений як посилання. Можна виділити три типи гіперпосилань: внутрішні, в межах одного документу та зовнішні.

Внутрішніми називають посилання які пов'язують HTML-документи (сторінки) в один сайт і допомагають пересуватися по ньому.

Гіперпосилання в межах одного документа організують переходи всередині одного HTML-документа. Вони застосовуються, коли на одній сторінці багато тексту. Щоб створити таке посилання, спочатку потрібно визначити місце - мітку, до якого посилання приводить. Потім потрібно створити посилання на цю мітку.

Зовнішніми називають посилання на об'єкти, розташовані поза межами поточного Web-сайту. Основою зовнішніх посилань є URL-адреса об'єкта, на який йде посилання.

Будь-який тип посилання можна реалізувати у вигляді зображення, тобто нова сторінка буде відкриватися при клацанні кнопкою миші на малюнку. Для створення посилання-зображення достатньо помістити малюнок всередині елемента **A**.

Основні теги та їх атрибути призначені для створення гіперпосилань наведені у додатку 1.

Порядок виконання роботи:

1. Об'єднати посиланнями Web-сторінки, які отримані в результаті виконання лабораторної роботи №3, у відповідності до розробленої навігаційної схеми Web-вузла (див. лабораторну роботу №2). При цьому передбачити внутрішні, зовнішні посилання та посилання в межах одного документу.

2. Провести перевірку функціонування гіпертекстових посилань у браузері та впевнитись у вірності переходів за ними. У разі необхідності відкоригувати гіперпосилання, скориставшись переглядом змісту сторінки у вигляді html-коду.

3. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Розкрийте сутність гіперпосилання.

2. Які типи гіперпосилань ви знаєте?

3. Яким чином задаються гіпертекстові посилання між окремими сторінками одного сайту?

4. Яким чином задаються гіпертекстові посилання в межах одного документу?

5. Яким чином задаються гіпертекстові посилання на інші сайти?

6. Наведіть приклади посилань на інші об'єкти.

7. Як задати URL-адресу на файл власного комп'ютера?

Тема 5. Графіка в гіпертекстових документах – 4 год.

Лабораторна робота №5. Розміщення та відтворення графіки на Web-сторінках

Мета роботи: набути практичних навичок роботи з графічними елементами Web-сторінок.

Загальні положення.

Робота з зображеннями і графікою в гіпертекстових документах дуже важлива, саме вона дозволяє зробити зовнішній вигляд сторінки таким, яким ми його і хочемо бачити. Графіку та різноманітну анімацію використовують також для залучення нових клієнтів, створення помітного і унікального дизайну для сайту. Для вставки графічного зображення застосовується тег ****. На даний час підтримуються формати файлів **.gif, .jpg, .jpeg, .png**.

Не варто забувати, що графіка повинна гармонійно поєднуватися із загальним настроєм сайту. Вона не повинна різко виділятися, а лише трохи підкреслювати тематику сайту. Не можна використовувати яскраві, кислотні кольори, які різко контрастують з кольором тексту, не можна використовувати фонові кольори, близькі до кольору тексту, не можна перевантажувати сторінку великою кількістю анімації – все це відлякує користувача і збільшує час завантаження сторінки.

Для використання на гіпертекстових сторінках, графіка повинна бути підготовлена заздалегідь в будь-якому графічному редакторі, отримана за допомогою цифрової камери, сканера, або завантажена з мережі Інтернет.

Основні теги та їх атрибути призначені для роботи з графікою наведені у додатку 1, а базові кольори представлені у додатку 2.

Порядок виконання роботи:

1. Розмістити графічні елементи (малюнки, фотографії, візуальні елементи навігації тощо) на раніше створених Web-сторінках у відповідності із плануванням вмісту Web-вузла та Web-сторінок (див. лабораторні роботи №2 та №3). Передбачити можливість відображення текстового повідомлення при неможливості відобразити малюнок. Задати розташування та розмір графічних елементів. Там де це потрібно, задати фонове зображення сторінки.

2. Провести перевірку місця розташування та коректності відображення графіки на Web-сторінках за допомогою браузера. У разі необхідності відкоригувати Web-сторінки, скориставшись переглядом їх змісту у вигляді html-коду.

3. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Які графічні зображення можна розміщувати в гіпертекстових документах?

2. Опишіть процедуру вставки графічних зображень у гіпертекстові документи.
3. Які способи вирівнювання графічних зображень ви знаєте?
4. Яким чином задається висота та ширина графічних зображень?
5. Яким чином задається рамка навколо графічного зображення?
6. Як задати альтернативний текст, що виводиться замість зображення?
6. Як змінити відступи між графічним зображенням та текстом?
7. Що потрібно для того, щоб перетворити графічне зображення у гіперпосилання?
8. Як встановити фонове зображення у гіпертекстовому документі?

Тема 6. Таблиці в гіпертекстових документах – 4 год.

Лабораторна робота №6. Створення Web-сторінок за допомогою таблиць

Мета роботи: набути практичних навичок зі створення та розміщення таблиць на Web-сторінках, побудови Web-сайтів за їх допомогою.

Загальні положення.

У технічній та економічній літературі, в різних документах таблиці використовують, для того, щоб розташувати інформацію в простому і зрозумілому вигляді. У мові HTML таблиці використовуються в двох випадках:

1. для представлення числових та інших даних, розбитих за рядками і стовпцями;
2. як засіб форматування гіпертекстових сторінок, завдання взаємного розташування елементів сторінки.

У другому випадку комірки таблиці можуть містити будь-які HTML-елементи, наприклад: заголовки, списки, абзаци, фігури, графіку, а також елементи форм. Фактично весь сайт розташовується у великій складній таблиці.

Опис таблиць повинен розташовуватися усередині елемента **BODY**. Документ може містити довільну кількість таблиць, допускається вкладення таблиць одна в одну. Кожна таблиця повинна починатися тегом **<TABLE>**, а і завершуватися тегом **</TABLE>**. Всі інші елементи таблиці повинні бути вкладеними в елемент **TABLE**.

Виділяють наступні елементи таблиці:

Комірка – це основний елемент таблиці. Вона формується перетином рядка і стовпця.

Рядок – це пряма лінія комірок, розташованих зліва направо.

Стовпець – це набір комірок, розташованих одна над іншою зверху вниз.

Рамка – це лінія, яка оточує кожну комірку і таблицю в цілому.

Основні теги та їх атрибути призначені для роботи з таблицями наведені у додатку 1, а базові кольори представлені у додатку 2.

Порядок виконання роботи:

1. Розмістити таблиці на раніше створених Web-сторінках у відповідності із плануванням вмісту Web-вузла та Web-сторінок (див. лабораторні роботи №2 та №3). Задати вирівнювання таблиць, їх розміри, колір.

2. Використати невидимі таблиці в якості засобу форматування Web-сторінок, завдання взаємного розташування графічних елементів сторінки.

3. Провести перевірку відображення таблиць на гіпертекстових сторінках за допомогою браузеру. У разі необхідності відкоригувати гіпертекстові сторінки, скориставшись переглядом їх змісту у вигляді html-коду.

4. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. З яких елементів складається таблиця?
2. Як визначити структуру таблиці?
3. Яким чином задається заголовок таблиці?
4. За допомогою яких атрибутів встановлюється рамка таблиці?
5. Як змінити відступи між рамкою таблиці та текстом?
6. Які способи вирівнювання елементів таблиці ви знаєте?

7. Яким чином задається висота та ширина таблиці?
8. Як встановити фонові колір або малюнок таблиці?
9. Для чого і яким чином виконується групування рядків та стовпчиків таблиці?
10. Наведіть метод вкладення однієї таблиці в іншу.

Тема 7. Каскадні таблиці стилів CSS (Cascading Style Sheets) – 2 год.

Лабораторна робота №7. Стилізація Web-сторінок за допомогою каскадних таблиць стилів

Мета роботи: ознайомитись з можливостями каскадних таблиць стилів та оволодіти навичками стилізації Web-сторінок за їх допомогою.

Загальні положення.

Стандарти листів стилів для Web були розроблені консорціумом W3C у 1995–1996 рр. і названі **Cascading Style Sheets** (каскадні листи стилів), або скорочено – **CSS**. Присутність слова “каскадні” в цьому терміні означає, що листи стилів дозволяють створювати ієрархію стильових властивостей, згідно з якою локальний стиль скасовує глобальний стиль.

При стилізації гіпертекстового документа за допомогою каскадних листів стилів він представляється як структура вкладених елементів різного рівня, в якій стиль елемента більш низького рівня має пріоритет перед стилем зовнішнього елемента високого рівня.

Листи стилів – це фактично набори CSS-правил, які задають властивості форматування елементів документа.

До **внутрішніх листів стилів** згідно специфікації CSS відносяться вбудовані та заголовні стилі. Ця назва вказує на те, що визначення стилів виконується в рамках одного документа.

Вбудований стиль – це спосіб форматування елементів HTML через атрибут **style**.

Заголовний стиль – це розміщення листа стилів в головній частині документа (в елементі **HEAD**). Відповідно до цього способу можна єдиним чином управляти представленням всього документа.

Для розміщення інформації про стиль застосовується спеціальний контейнер **<STYLE>**
</STYLE>, який вставляється в заголовну частину документа.

Зовнішні листи стилів записуються в окремих файлах і застосовуються для оформлення набору гіпертекстових документів. Використання зовнішніх аркушів стилів дозволяє єдиним чином оформляти безліч Web-сторінок і навіть сайтів. Опис стилів зберігається в окремому файлі, який має розширення **.css** або **.jss**.

Основним інструментом підключення до гіпертекстового документу зовнішніх аркушів стилів є одиночний тег **<LINK>**. Цей тег розташовується в заголовній частині **HEAD**.

Основні теги та їх атрибути призначені для роботи з каскадними таблицями стилів наведені у додатку 1.

Порядок виконання роботи:

1. Виконати форматування раніше створених Web-сторінок за допомогою каскадних таблиць стилів. Для цього сформувати зовнішній лист стилів **mystyle.css** використовуючи не менше п'яти елементів із заданими визначеннями стилю.

2. Підключити сформований лист стилів до Web-сторінок в яких він буде використовуватися.

3. Провести перевірку відображення Web-сторінок за допомогою браузеру. У разі необхідності відкоригувати Web-сторінки, скориставшись переглядом їх змісту у вигляді **html-коду**.

4. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Розкрийте поняття каскадних аркушів стилів.
2. Що називають стильовими властивостями?
3. Які можливості та переваги надає використання CSS?

4. Дайте визначення вбудованого стилю.
5. Яким чином формується аркуш стилів?
6. Що таке визначення стилю, селектор?
7. Наведіть види селекторів.
8. Як формуються зовнішні аркуші стилів?
9. Наведіть відомі вам стильові властивості.

Тема 8. Фрейми – 4 год.

Лабораторна робота №8. Розробка Web-сайтів із використанням фреймів

Мета роботи: ознайомитися із поняттям фреймів та навчитися створювати Web-сайти за їх допомогою.

Загальні положення.

Фрейми (frames – кадр, рамка) дозволяють відкрити у вікні браузера не один, а відразу декілька гіпертекстових документів. Фрейми використовують для втілення сміливих дизайнерських задумів, у випадках, коли їх неможливо уникнути, або тому, що з їхньою допомогою можна швидко і зручно створити повноцінний Web-сайт.

Фрейм-документ створюється за допомогою елемента **FRAMESET**, який замінює собою звичний елемент **BODY**.

Існує багато варіантів розташування фреймів, проте найбільш уживаними є чотири варіанти:



Основні теги та їх атрибути призначені для роботи з фреймами наведені у додатку 1.

Порядок виконання роботи:

1. Використовуючи фрейми (розбиття вікна браузера на кадри) та раніше розроблені Web-сторінки створити тематично та функціонально завершений Web-сайт у відповідності до розробленої навігаційної схеми Web-вузла (див. лабораторну роботу №2).

2. Провести перевірку функціонування гіпертекстових посилань отриманого Web-сайту у браузері та впевнитись у вірності переходів за ними. У разі необхідності відкоригувати гіперпосилання, скориставшись переглядом змісту сторінки у вигляді html-коду.

3. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Розкрийте поняття “фрейм”.
2. Наведіть відомі вам типові структури фреймових документів.
3. За допомогою якого тегу створюється фреймовий документ?
4. За допомогою якого тегу задаються фреймові рядки та стовпчики?
5. Яким чином встановлюються розміри фреймів?
6. Як змінити поля фреймів?
7. Для чого може виникнути необхідність відключення скролінгу по фрейму?
8. Яким чином треба задавати гіперпосилання для відкриття документа в потрібному фреймі?

Тема 9. Реалізація клієнтської активності – 2 год.

Лабораторна робота №9. Реалізація клієнтської активності з використанням форм

Мета роботи: набути практичних навичок реалізації клієнтської активності з використанням форм.

Загальні положення.

Форми (Web-форми) призначені для пересилання даних від віддаленого користувача до Web-серверу. З їх допомогою можна організувати найпростіший діалог між користувачем і

сервером, наприклад, реєстрацію користувача на сервері або вибір потрібного документа з представленого списку. Форми підтримуються всіма популярними браузерами.

На Web-сайтах для завдання форми використовують елемент **FORM**. Документ може містити декілька форм, але вони не можуть бути вкладеними.

Порядок виконання роботи:

1. Застосувати різноманітні форми для реалізації клієнтської активності на розробленому Web-сайті. Передбачити Web-форми для введення та відсилки даних, поля для встановлення прапорців (перемикачів) або меню вибору варіанту, різноманітні кнопки. Реалізувати відсилку заповненої форми на адресу електронної пошти.

2. Провести перевірку функціонування впроваджених на Web-сайті форм. У разі необхідності відкоригувати Web-сторінки, скориставшись переглядом змісту сторінки у вигляді html-коду.

3. За результатами роботи скласти письмовий звіт.

Питання до самоконтролю:

1. Розкрийте поняття “форма”.
2. Чим відрізняються методи пересилки інформації GET та POST?
3. За допомогою якого тега створюються поля для вводу рядків тексту, пароля, імені локального файлу?
4. Для чого потрібні такі елементи як прапорці, перемикачі та яким чином вони створюються?
5. За допомогою яких тегів створюються кнопки для пересилки інформації та скидання?
6. За допомогою яких тегів створюються елементи-графічні зображення?
7. Для чого потрібні приховані елементи та як вони створюються?
8. Що таке випадajući списки та як вони створюються?

3. ОРІЄНТОВНА ТЕМАТИКА РЕФЕРАТИВ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Життєвий цикл Web-ресурсу та зміст його етапів
2. Визначення аудиторії Web-сайтів та її характеристика
3. Інструменти роботи з аудиторією Web-сайту
4. Інформаційне наповнення та підтримка Web-сайтів
5. Засоби процесу розробки та експлуатації Web-сайтів
6. Адміністративна частина керування Web-сайтом
7. Служби взаємодії з Web
8. Шифрування даних Web-серверами
9. Інструментальні засоби підготовки текстів для Інтернет
10. Огляд систем управління контентом (CMS)
11. Тестування Web-сайтів
12. Організація служби банерного обміну та банерні системи
13. Таргетинг, критерії вибору банерних мереж
14. Принципи розміщення реклами у пошукових системах
15. Особливості функціонування пошукових систем і методи пошукової оптимізації
16. Сучасні технології створення Web-сайтів
17. Огляд програмних продуктів для створення Web-сайтів
18. Переваги і недоліки модульного Web-програмування
19. Роль динамічного програмування при створенні Web-сторінок
20. Мови розробки гіпертекстових сторінок
21. Мови динамічної Web-розробки
22. Мови сценаріїв
23. Мова активних сторінок сервера Microsoft ASP (Active Server Pages)
24. Різновиди середовищ PERL (Practical Extraction and Report Language)
25. Стандартна узагальнена мова розмітки (SGML)

26. Огляд мови програмування PHP (Hypertext Preprocessor)
27. Особливості об'єктно-орієнтованої скриптової мови програмування JavaScript
28. Особливості об'єктно-орієнтованої мови програмування Java
29. Основні можливості та відмінності мови HTML5 (HyperText Markup Language, ver. 5)
30. Інтерфейс програмування додатків XHR (XMLHTTP, XMLHttpRequest)
31. Мова математичної розмітки MathML (Mathematical Markup Language)
32. Мова розмітки інтерактивних мультимедійних презентацій SMIL (The Synchronized Multimedia Integration Language)

Загальні вимоги до рефератів

Реферат є найпростішою формою наукового дослідження, мета якого полягає в ознайомленні студентів з навчальною літературою, виробленні умінь, необхідних для здійснення бібліографічного пошуку, оволодіння науковим, науково-популярним стилем викладу засвоєної інформації у вигляді стислої усної або письмової доповіді. В подальшому це має допомогти студентам у написанні тез доповідей на конференції, статей, курсових та дипломних робіт.

Реферати пишуться на аркушах формату А4, загальний обсяг 12-15 сторінок рукописного або друкованого тексту. Титульна сторінка включає такі компоненти: назву міністерства, навчального закладу, факультету, тему реферату, назву навчальної дисципліни, прізвище і групу студента, який писав реферат, прізвище викладача, який має перевірити дану роботу. На наступній сторінці подається план реферату за простою або складною формою. Зміст реферату обов'язково включає три компоненти: вступ, основна частина, висновки.

У вступі розкривається актуальність теми, висвітлюються питання історіографії; в основній частині студент демонструє володіння матеріалом, уміння його компонувати відповідно до поставленої мети, уміння логічно висловлювати свої думки; у висновках підсумовуються результати проведених студентом досліджень, які можуть подаватися довільно.

Реферат завершує список використаної літератури, оформлений відповідно до вимог стандартів: ДСТУ ГОСТ 7.1:2006 "Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання (ГОСТ 7.1–2003, IDT)". Джерела необхідно розмішувати у порядку появи посилань у тексті.

4. КОНТРОЛЬНІ ПИТАННЯ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Назвіть мету та завдання дисципліни "Web-програмування".
2. Розкрийте поняття всесвітньої мережі Інтернет.
3. Назвіть основні сервіси мережі Інтернет.
4. Що таке Web-браузер та які вони бувають?
5. Що таке Web-сервер та для чого він потрібен?
6. Розкрийте поняття гіпертексту.
7. Розкрийте сутність служби WWW (World Wide Web).
8. Які види Web-сторінок ви знаєте?
9. Охарактеризуйте основні типи Web-вузлів.
10. Назвіть та опишіть етапи планування Web-вузлів.
11. Які загальні правила проектування навігаційної схеми сайту ви знаєте?
12. Розкрийте ролі у Web-дизайні.
13. Дайте визначення HTML (HyperText Markup Language).
14. Що називають розміткою документу?
15. Надати основні відомості з історії розвитку мови HTML.
16. Охарактеризувати тегову модель мови HTML.
17. Розкрийте поняття парного, одиночного тега, контейнера.
18. Що називають елементами HTML?
19. Навести класифікацію елементів HTML.
20. Що таке вкладені елементи HTML?

21. Наведіть правила вкладення елементів HTML.
22. Що називають атрибутами тегів та які правила їх запису?
23. Наведіть основні правила синтаксису HTML.
24. Опишіть загальну структуру HTML-документа.
25. Що називають МЕТА-даними та яку роль для пошуку вони відіграють?
26. Розкрийте базові правила оформлення тексту в HTML-документах.
27. Наведіть основні елементи для форматування тексту.
28. Наведіть елементи для форматування великих блоків тексту.
29. Розкрийте сутність заголовків та методи їх створення.
30. Опишіть процедуру створення абзацу в гіпертекстовому документі.
31. Яким чином виконується примусове перенесення тексту на новий рядок?
32. Яким чином можна заборонити перенесення тексту на новий рядок?
33. Що таке списки та які вони бувають?
34. Опишіть процедуру створення маркованого списку.
35. Опишіть процедуру створення нумерованого списку.
36. Опишіть процедуру створення списку визначень.
37. Яким чином здійснюється управління шрифтом в HTML-документах?
38. Що таке базовий шрифт та яким чином він задається?
39. Яким чином реалізується прокручування тексту?
40. Як задати коментар до фрагменту коду в HTML-документі?
41. Розкрийте сутність гіперпосилання.
42. Які типи гіперпосилань ви знаєте?
43. Яким чином створюється гіперпосилання?
44. Опишіть створення внутрішніх гіперпосилань.
45. Опишіть особливості створення гіперпосилань в межах одного документу.
46. Опишіть створення зовнішніх гіперпосилань.
47. Наведіть приклади посилань на інші об'єкти.
48. Які графічні зображення можна розміщувати в гіпертекстових документах?
49. Опишіть процедуру вставки графічних зображень у гіпертекстові документи.
50. Які способи вирівнювання графічних зображень ви знаєте?
51. Яким чином задається висота та ширина графічних зображень?
52. Яким чином задається рамка навколо графічного зображення?
53. Як задати альтернативний текст, що виводиться замість зображення?
54. Як змінити відступи між графічним зображенням та текстом?
55. Що потрібно для того, щоб перетворити графічне зображення у гіперпосилання?
56. Наведіть спосіб встановлення графічного зображення у якості фону гіпертекстового документу.
57. У яких випадках в гіпертекстових документах використовуються таблиці?
58. З яких елементів складається таблиця?
59. Як визначити структуру таблиці?
60. Яким чином задається заголовок таблиці?
61. За допомогою яких атрибутів встановлюється рамка таблиці?
62. Як змінити відступи між рамкою таблиці та текстом?
63. Які способи вирівнювання елементів таблиці ви знаєте?
64. Яким чином задається висота та ширина таблиці?
65. Як встановити фонові колір або малюнок таблиці?
66. Для чого і яким чином виконується групування рядків та стовпчиків таблиці?
67. Наведіть метод вкладення однієї таблиці в іншу.
68. Розкрийте поняття каскадних аркушів стилів.
69. Що називають стильовими властивостями?
70. Які можливості та переваги надає використання CSS?
71. Дайте визначення вбудованого стилю.

72. Яким чином формується аркуш стилів?
73. Що таке визначення стилю, селектор?
74. Наведіть види селекторів.
75. Як формуються зовнішні аркуші стилів?
76. Наведіть відомі вам стильові властивості.
77. Розкрийте поняття “фрейм”.
78. Наведіть відомі вам типові структури фреймових документів.
79. За допомогою якого тегу створюється фреймовий документ?
80. За допомогою якого тегу задаються фреймові рядки та стовпчики?
81. Яким чином встановлюються розміри фреймів?
82. Як змінити поля фреймів?
83. Для чого може виникнути необхідність відключення скролінгу по фрейму?
84. Яким чином треба задавати гіперпосилання для відкриття документа в потрібному фреймі?
85. Розкрийте поняття “форма”.
86. Чим відрізняються методи пересилки інформації GET та POST?
87. За допомогою якого тега створюються поля для вводу рядків тексту, пароля, імені локального файлу?
88. Для чого потрібні такі елементи як прапорці, перемикачі та яким чином вони створюються?
89. За допомогою яких тегів створюються кнопки для пересилки інформації та скидання?
90. За допомогою яких тегів створюються елементи-графічні зображення?
91. Для чого потрібні приховані елементи та як вони створюються?
92. Що таке випадajući списки та як вони створюються?

5. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бесплатная Web-страница своими руками. Подробное иллюстрированное руководство ; под ред. В.Б. Комягина. – М. : Лучшие книги, 2005. – 237 с.
2. Богомолова О.Б. Web-конструирование на HTML : практикум / О.Б. Богомолова. – М. : БИНОМ. Лаборатория знаний, 2008. – 192 с.
3. Борисенко А.А. Web-дизайн. Просто как дважды два / А.А. Борисенко. – М. : Эксмо, 2008. – 320 с.
4. Внутренний мир World Wide Web ; пер. с англ. – 2-е изд. – К. : Dia Soft, 1997. – 544 с.
5. Гаевский А.Ю. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. – М. : ЭКСМО, 2008. – 454 с.
6. Использование World Wide Web / Д. Уолл, К. Купер, Д. Гиббонс. – 2-е изд. – К. ; М. : Диалектика, 1997. – 430 с.
7. Бабушкин М. Web-сервер в действии / М. Бабушкин, С. Иваненко, В. Коростелев. – СПб. : Питер, 1997. – 264 с.
8. Глинський Я.М. Інтернет. Сервіси, HTML і Web-дизайн : навч. посіб. / Я.М. Глинський, В.А. Ряжська. – 2-ге вид., доп. – Львів : Деол, 2003. – 190 с.
9. Квинт И. HTML, XHTML и CSS на 100 % / И. Квинт. – СПб. : Питер, 2010. – 384 с.
10. Пасічник О.Г. Основи веб-дизайну / О.Г. Пасічник, О.В. Пасічник, І.В. Стеценко : [навч. посіб.]. – К. : Вид. група ВНУ, 2009. – 336 с.
11. Перри П.Д. Секреты World Wide Web / П.Д. Перри. – К. : Диалектика, 1996. – 576 с.
12. Рева О.Н. Создание Web-страниц в кармане / О.Н. Рева. – М. : ЭКСМО, 2008. – 255 с.
13. Кирсанов Д. Веб-дизайн / Дмитрий Кирсанов. – СПб. : «Символ-Плюс», 2001. – 376 с.
14. Матросов А. HTML 4.0 / А. Матросов, А. Сергеев. – СПб. : «ВНУ», 2000. – 670 с.
15. Ковалев А. Управление проектом по созданию Интернет-сайта / Ковалев А., Курдюмов И. – М. : Альпина Паблишер, 2001. – 337 с.
16. Пауэлл Т.А. Полное руководство по HTML / Т.А. Пауэлл. – Мн. : ООО «Попурри», 2001. – 911 с.

17. Коржинский С.Н. Настольная книга Web-мастера: эффективное применение HTML, CSS и JAVASCRIPT / С.Н. Коржинский. Издательский торговый дом «КноРус», 2000. – 320 с.
18. Хольцшлаг Э.М. Использование HTML 4 / Хольцшлаг Э. Молли. – М.: Изд. дом «Вильямс», 2000. – 1008 с.
19. Кан Л. Мой узел Web / Луис Кан, Лаура Логан. – М. : Русская редакция, 1996. – 209 с.
20. Леонтьев Б. Web-дизайн: Тонкости, хитрости и секреты / Б. Леонтьев. – М. : Познавательная книга плюс, 1999. – 190 с.
21. Шафран Э. Создание WEB-страниц: Самоучитель / Э. Шафран. – СПб. : Питер, 1999. – 310 с.
22. Бланк-Эдельман Д. PERL для системного администрирования / Д. Бланк-Эдельман. – СПб. : Символ-Плюс, 2001. – 496 с.
23. Вивек Ш. Разработка Web-серверов для электронной коммерции. Комплексный подход / Шарма Вивек, Шарма Раджив. +CD. – М. : Изд. дом «Вильямс», 2001. – 400 с.
24. Грибов Д.Е. Macromedia Flash 4. Интерактивная Web-анимация / Д.Е. Грибов. – М. : ДМК, 2000. – 672 с.
25. Кастаньетто Дж. Профессиональное PHP программирование / Джезус Кастаньетто, Хариш Рават, Саша Шуман, Крис Сколло, Дипак Велиаф. – СПб. : Символ-Плюс, 2001. – 912 с.
26. Нидерст Дж. WEB-мастеринг для профессионалов / Дженнифер Нидерст. – СПб. : Питер, 2000. – 576 с.
27. Бранденбау Джерри. JavaScript: сборник рецептов / Джерри Бранденбау. – СПб. : Питер, 2000. – 416 с.
28. Либерти Дж. Создание документов XML для Web на примерах / Джесс Либерти, Майк Крейли. – М. : Изд. дом «Вильямс», 2000. – 256 с.
29. Мильберн П. Кен. FLASH 5 для дизайнера / П. Кен Мильберн, Джон Крото. +CD. – К. : Diasoft, 2000. – 496 с.
30. Мук К. ActionScript. Подробное руководство / Колин Мук. – СПб. : Символ-Плюс, 2002. – 792 с.
31. Заккар Меградж. Разработка приложений для электронной коммерции на Oracle8i и Java / Меградж Заккар. – М. : Изд. дом «Вильямс», 2000. – 336 с.
32. Монкур Майкл. Освой самостоятельно JavaScript 1.3 за 24 часа / Майкл Конкур. – М. : Изд. дом «Вильямс», 2000. – 304 с.
33. Питц-Моултис Н. XML в подлиннике. Для широкого круга Web-разработчиков / Н. Питц-Моултис, Ч. Кирк. – СПб. : «ВНУ-Санкт-Петербург», 2000. – 736 с.
34. Чейз Н. Active Server Pages 3.0 на примерах / Николас Чей. – М. : Изд. дом «Вильямс», 2001. – 352 с.
35. Пауелл Томас. Web-дизайн / Томас Пауелл. – СПб. : «ВНУ Петербург», 2002. – 1024 с.
36. Ноутон П. JAVA 2. В подлиннике. Наиболее полное руководство / П. Ноутон, Г. Шилдт. – СПб. : «ВНУ-Санкт-Петербург», 2000. – 1072 с.
37. Уэйнрайт Питер. Apache для профессионалов / Питер Уэйнрайт. – М. : «Лори», 2001. – 474 с.
38. Спенсер Пол. XML. Проектирование и реализация / Пол Спенсер. – М. : «Лори», 2001. – 509 с.
39. Круг Стив. Веб-дизайн: книга Стива Круга, или «не заставляйте меня думать!» / Стив Круг. – СПб. : Символ-Плюс, 2001. – 200 с.
40. Кристиансен Т. PERL: библиотека программиста / Том Кристиансен, Натан Торкингтон. – СПб. : Питер, 2001. – 736 с.
41. Фролов А.В. Базы данных в Интернет. Практическое руководство по созданию Web-приложений / А.В. Фролов. – М. : Русская редакция, 2000. – 448 с.
42. Холл М. Программирование для Web. Библиотека для профессионалов / М. Холл. – М. : Изд. дом «Вильямс», 2002. – 1216 с.
43. Швендимен Блейк. PHP 4. Руководство разработчика / Швендимен Блейк. – М. : Изд. дом «Вильямс», 2002. – 684 с.
44. Нильсен Якоб. Веб-дизайн. Книга Якоба Нильсена / Якоб Нильсен. – СПб. : Символ-Плюс, 2001. – 512 с.
45. Вуд Л. Web-графика : справочник / Л. Вуд. – СПб. : Питер, 1998. – 217 с.
46. Нидерст Дж. Web-мастеринг для профессионалов: Настольный справочник / Джон Нидерст. – СПб. : Питер, 2001. – 569 с.

47. Григин И.Е. РНР 4. Специальный справочник / И.Е. Григин. – СПб. : Питер, 2002. – 672 с.

Основні теги мови HTML

Тег		Опис тега	
Структурні теги мови HTML			
<HTML>... </HTML>		Обмежує HTML-документ	
<HEAD>... </HEAD>		Задає заголовок тексту	
<TITLE>... </TITLE>		Розміщується в заголовку, містить назву сторінки	
Теги для роботи з текстом			
<H1> заголовок 1 </H1>		Заголовок найбільшого розміру	
<H6> заголовок 6 </H6>		Заголовок найменшого розміру	
<PRE>... </PRE>		Обрамлення попередньо оформленого тексту	
... 		Напівжирний шрифт	
<I>... </I>		Курсив	
<U>... </U>		Підкреслення	
 		Перенесення наступного тексту на новий рядок (кінець рядка)	
<P>		Повідомляє браузеру про початок нового абзацу (на відміну від попередньої команди ще й вставляє порожній рядок між абзацами)	
<BLOCKQUOTE>... </BLOCKQUOTE>		Створює блок тексту з відступом	
<HR>		Вставка горизонтальної лінії на всю ширину екрану	
<CENTER>... </CENTER>		Центрування блоку тексту	
<!-- коментар -->		Задає коментар	
Теги для створення списків і відступів			
... ... 		Задає відступ – текст належить зсунути вправо на 1 крок табуляції (певне число символів)	
 елемент 1 елемент 2 		Задає невпорядкований список елементів (елементи позначаються спеціальними символами)	
 елемент 1 елемент 2 		Задає впорядкований список нумерованих елементів	
<DL> <DT> термін </DT> <DD> визначення </DL>		Задає списки-визначення	
<DIR> ... </DIR>		Непомічені списки	
Теги для створення гіпертекстових посилань			
 анкер 		Задає посилання на інший файл на іншій Web-сторінці. Анкером може бути текст або зображення (використовується команда вставки зображення). Анкер відображається браузером у документі, і у разі його вибору здійснюється перехід за посиланням	
 анкер 		Задає посилання для виклику іншого файлу з того ж самого ПК	
 анкер 		Задає посилання на місце всередині цього ж файлу. Аби позначити місце, куди буде здійснено перехід, використовується команда: ...	
Теги з параметрами			
Тег	Опис тега	Параметри	Опис параметра
<BODY>... </BODY>	Обмежує тіло документа	background="ім'я файлу" bgcolor="#RRGGBB" text="#RRGGBB" link="#RRGGBB" alink="#RRGGBB" vlink="#RRGGBB"	Задає файл зі "шпалерами" Колір тла Колір тексту Колір невикористаних гіпертекстових посилань Колір активних посилань Колір використаних посилань
... 	Визначає шрифт тексту	color="#RRGGBB" face="ім'я шрифту",... "ім'я шрифту" size=1...7	Колір шрифту Список імен шрифтів, одним із яких відображатиметься текст Розмір шрифту (1 – найменший)
<P>...	Вирізняє	align="вирівнювання"	Задає вирівнювання абзацу

</P>	абзац		відносно меж документа: left, right, center
	Перегляд графічних зображень	src="ім'я файлу" alt="опис" align="вирівнювання" vspace=число hspace=число height=число width=число	Файл із зображенням у форматі .gif, .jpg або .png Альтернативний текст – опис картини для текстового режиму Розміщення зображення: left, right, center, top, bottom, middle Відступ від тексту по вертикалі Відступ від тексту по горизонталі Видима висота зображення Видима ширина зображення
<TABLE> </TABLE>	Задає таблицю	border=число cellspacing=число cellpadding=число width=число (відсоток) height=число (відсоток) align="вирівнювання" bgcolor="#RRGGBB" bordercolor="#RRGGBB"	Ширина рамки таблиці Товщина розділювачів Відступ від елемента до лінії Ширина таблиці в пікселях або % від ширини вікна Висота таблиці в пікселях або % Місце таблиці в тексті: left, right, center Колір фону комірок таблиці Колір рамки таблиці
<CAPTION>... </CAPTION>	Задає назву таблиці	align="вирівнювання"	Визначає місце назви відносно таблиці: top, bottom
<TR>... </TR>	Задає рядок таблиці	align="вирівнювання" valign="вирівнювання"	Вирівнювання тексту по горизонталі: left, right, center Вирівнювання тексту по вертикалі: top, bottom, middle
<TD>... </TD> <TH>... </TH>	Задає комірку (TD) або комірку-заголовок (TH) у поточно-му рядку таблиці	align="вирівнювання" valign="вирівнювання" width=число (відсоток) height=число (відсоток) colspan=число rowspan=число nowrap	Вирівнювання тексту по горизонталі: left, right, center Вирівнювання тексту по вертикалі: top, bottom, middle Ширина комірки Висота комірки Кількість стовпців у комірці Кількість рядків у комірці Блокує автоматичне перенесення слів в межах поточної комірки
<FRAMESET>... </FRAMESET>	Задає фрейм-документ	rows=число (відсоток) cols=число (відсоток) border=число	Висота фрейму в пікселях або % від висоти вікна Ширина фрейму в пікселях або % від ширини вікна Ширина рамки між фреймами
<FRAME>	Визначає вигляд та зміст фрейму	src="ім'я файлу" noresize scrolling="тип скролінгу" marginheight=число marginwidth=число	Визначає документ, що буде міститися у фреймі Блокує можливість зміни розміру фрейму Задає управління скролінгом: no, yes, auto Ширина верхнього та нижнього полів фрейму в пікселях Ширина лівого та правого полів фрейму в пікселях
<FORM>...</FORM>	Задає форми	action="URL" method enctype	Програма, яка буде обробляти інформацію з форми Метод пересилки: GET, POST Визначає формат кодування даних при їх передачі

Назви базових кольорів

Ім'я кольору	Опис	Код кольору у форматі RRGGBB (шістнадцяткове значення)
aqua	Блакитний	#00ffff
black	Чорний	#000000
blue	Синій	#0000ff
fuchsia	Фуксія	#ff00ff
gray	Сірий	#808080
green	Зелений	#008000
lime	Світло-зелений	#00ff00
maroon	Темно-червоний	#800000
navy	Темно-синій	#000080
olive	Оливковий	#808000
purple	Фіолетовий	#800080
red	Червоний	#ff0000
silver	Світло-сірий	#c0c0c0
teal	Синьо-зелений	#008080
white	Білий	#ffffff
yellow	Жовтий	#ffff00

ПРИМІТКИ

ПРИМІТКИ