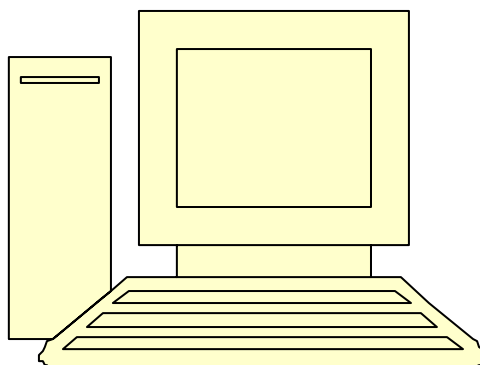


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА
ЦЕНТР ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА КОМП'ЮТЕРНИХ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних робіт на тему «Основи Web-дизайну»
для студентів центру післядипломної освіти
всіх спеціальностей та форм навчання



Полтава 2008

Методичні вказівки для проведення лабораторних робіт на тему «Основи web-дизайну» для студентів центру післядипломної освіти всіх спеціальностей та форм навчання. – Полтава: ПолтНТУ, 2008. – 29 с.

Укладачі: Т.М.Деркач, канд.техн.наук, доцент, Т.А.Дмитренко, ст. викладач.

Відповідальний за випуск: завідувач кафедри комп'ютерних та інформаційних технологій і систем О.Л. Ляхов, доктор техн.наук, доцент.

Рецензенти: Г.В.Головко канд.техн.наук, доцент,
В.В.Васюта канд.техн.наук, доцент

Затверджено науково-методичною радою університету

Протокол № _____ від . _____

Редактор Я.В. Новічкова

Загальні теоретичні відомості	4
1. Рекомендації щодо проектування сайта.....	4
2. Основні поняття і правила мови HTML. Структурні елементи форматування	5
3. Основні теги форматування тексту	6
4. Теги для форматування шрифту	9
5. Теги створення списків	12
6. Таблиці HTML	14
7. Гіперпосилання	17
8. Робота з графікою.....	18
9. Фрейми	20
Завдання для лабораторних занять	25
Завдання 1. Простий документ HTML	25
Завдання 2. Документ HTML із посиланнями.....	25
Завдання 3. Створення списків	25
Завдання 4. Створення таблиць	26
Завдання 5. Робота з кольором і графікою	26
Завдання 6. Робота з формами	26
Завдання 7. Робота з фреймами	26
Контрольне завдання.....	27

Загальні теоретичні відомості

1. Рекомендації щодо проектування сайта

На даний час пошук ефективних методів Web-проектування ведеться у двох напрямках – інформаційний дизайн і практичний дизайн. Основне завдання Web-дизайнера – оформити всі Web-сторінки вузла в єдиному стилі: використовувати своєрідні малюнки, однакові навігаційні кнопки, маркери, лінії, кольори і навіть шрифти.

При розробленні Web-сторінок слід дотримуватися деяких стандартних рекомендацій:

- Червоний, жовтий і зелений – це найскладніші кольори для сприйняття тексту. Їх слід використовувати у виняткових випадках.
- Швидкість читання Web-сторінок практично вдвічі нижча, ніж швидкість читання сторінки звичайного друкованого матеріалу. Ураховуючи це співвідношення, треба намагатися, щоб Web-сторінки не були довгими (не більше двох-трьох екранів монітора по вертикалі). Між ємними текстовими блоками слід вставляти таблиці й маркіровані списки.
- Невеликі за об'ємом абзаци мають бути розділені відступами. Після кожного з абзців, об'єм яких не повинен перевищувати десяти рядків, необхідно зробити відступи. Текстові блоки мають бути не тільки змістовними, але й читабельними.
- Для оформлення текстових блоків не слід використовувати дуже багато різноманітних шрифтів. Із практичної точки зору достатньо трьох різних шрифтів: один – для заголовків, другий – для текстових блоків, третій – для специфічних цілей, наприклад для супровідного тексту.
- Не потрібно зловживати графічними зображеннями. Кожний малюнок повинен відповідати загальній тематиці Web-сторінки або певної статті, яку він ілюструє.
- Для того щоб Web-сторінки виглядали привабливими, слід по-різному задавати вирівнювання малюнків щодо текстових блоків.

Проектування сайта виконується у два етапи:

1. Планування сайта в цілому.
2. Планування окремих сторінок.

Планування сайта в цілому. Тут необхідно уявити, з яких розділів складатиметься сайт, які сторінки в нього входитимуть. Спочатку потрібно зробити головну сторінку, котра завантажуватиметься першою і буде розповідати відвідувачам про те, що це за сайт, про що тут йде мова і як знайти інформацію, що цікавить. Тут же потрібно продумати навігацію по сайту, тобто спланувати, як відвідувач переміщатиметься із сторінки на сторінку.

Планування окремих сторінок. Після того як складений план сайту в цілому, потрібно продумати, як виглядатимуть окремі сторінки. Формувати або верстати сторінку можна двома способами: за допомогою фреймів і за допомогою таблиць.

2. Основні поняття і правила мови HTML. Структурні елементи форматування

HTML призначений для розмітки текстових документів (тобто за допомогою його можна розмістити на сторінці текст, таблиці, картинки тощо). Те, як виглядатимуть картинки, текст та інші елементи на сторінці і як вони розташовуватимуться відносно один одного, визначають мітки (tags, або теги).

Теги бувають двох типів: **теги-контейнери** та **теги-команди**. **Теги-контейнери** складаються з початкового і кінцевого компонентів (парних тегів), між якими розміщуються текст та інші елементи документа. Ім'я кінцевого тега ідентичне імені початкового, але перед ім'ям кінцевого ставиться коса риса (/).

Основними елементами, які визначають структуру HTML-документа, є такі: `<HTML> </html>`, `<HEAD> </head>`, `<TITLE> </title>`, `<BODY> </body>`.

```
<HTML> < /html>
```

Будь-яка WEB-сторінка починається з тега `<HTML>` і закінчується тегом `</HTML>`:

```
<HTML> Зміст документа </HTML>.
```

Даний елемент є зовнішнім, тому що між його початковим і кінцевим тегами повинна знаходитися вся WEB-сторінка. Тег допускає вкладення інших тегів, що визначають загальну структуру WEB-сторінки: `HEAD`, `BODY`, `FRAMESET` та ін. Кінцевим тегом `</HTML>` закінчуються всі WEB-сторінки.

```
<HEAD> < /head>
```

Даний тег визначає область заголовка Web-сторінки. Іншими словами, її першу частину. Так само, як і попередній елемент, `HEAD` служить тільки для формування загальної структури документа. Цей елемент може мати атрибути `lang` і `dir`, повинний включати елемент `TITLE` і допускає вкладення елементів `BASE`, `META`, `LINK`, `OBJECT`, `SCRIPT`, `STYLE`.

```
<TITLE> < /title>
```

Елемент для розміщення заголовка Web-сторінки. Рядок тексту, розташований усередині цього елемента, відображається не в документі, а в заголовку вікна броузера. Цей рядок часто використовується при організації пошуку в WWW. Тому необхідно подбати про те, щоб цей рядок не був занадто довгим, але досить точно відображав призначення документа.

```
<STYLE> < /style>
```

Містить опис стилю деяких елементів Web-сторінки.

```
<META>
```

Цей елемент містить службову інформацію, що не відбивається при огляді Web-сторінки. Усередині нього немає тексту у звичайному розумінні, тому немає і кінцевого тега. Кожний елемент `META` містить два основних атрибути, перший з яких визначає тип даних, а другий – зміст.

```
<BODY> </body>
```

Такий елемент складається з тексту, що визначає власне Web-сторінку. Це та довільна частина документа, яку розробляє автор сторінки і яка відображається броузером. Відповідно кінцевий тег цього елемента треба шукати наприкінці HTML-файлу. Усередині елемента `BODY` можна використовувати всі елементи, призначені для розробки Web-сторінки. Усередині початкового тега елемента `BODY` можна розташувати ряд атрибутів, що забезпечують установки для всієї сторінки цілком.

Загальна структура типового документа HTML виглядає таким чином:

```
<HTML>  
<HEAD>  
<TITLE>  
Текст заголовка вікна  
</TITLE>  
</HEAD>  
<BODY>  
Текст документа  
</BODY>  
</HTML>
```

3. Основні теги форматування тексту

Тег-контейнер `<P>...</P>` – тег абзацу. Закриваючий елемент не обов'язковий, але, якщо він відсутній, броузер не залишає після абзацу порожній рядок.

Приклад класичного документа.

<pre><html> <head> <title>Моя перша Web-сторінка</title> </head> <body> <p> Це моя перша Web-сторінка</p> </body> </html></pre>	Це моя перша Web-сторінка
---	---------------------------

За умовчанням броузер зазвичай форматує абзац із вирівнюванням по лівому краю. Для примусового вирівнювання служить атрибут `align=`, який може приймати значення:

- `LEFT` – вирівнювання по лівому краю;
- `CENTER` – вирівнювання по центру;
- `RIGHT` – вирівнювання по правому краю;
- `JUSTIFY` – вирівнювання по ширині.

<pre> <html> <head> <title>Моя перша Web-сторінка</title> </head> <body> <p align="left">вирівнювання абзацу ліворуч...</p> <p align="right">вирівнювання абзацу праворуч...</p> <p align="center">вирівнювання абзацу по центру...</p> <p align="justify">вирівнювання абзацу по ширині...</p> </body> </html> </pre>	<p>вирівнювання абзацу ліворуч...</p> <p>вирівнювання абзацу праворуч...</p> <p>вирівнювання абзацу по центру...</p> <p>вирівнювання абзацу по ширині...</p>
--	--

Тег `
` забезпечує примусовий перехід на новий рядок. Він має тільки початковий тег. У місці його розміщення рядок закінчується, а текст, що залишився, друкується з нового рядка.

<pre> <html> <head> <title> Моя перша Web-сторінка. Вірш</title> </head> <body> <p> Я в ліс по проліски пішов. </br> Це не далеко від Полтави. </br> Що ж, проліски – моя любов. </br> Ще рання. Хай і не яскрава. </br></p> <p> Цвітуть на пагорбках вони, </br> Оточені кругом водою, </br> Як перші вісники весни </br> З їх радістю і їх журбою. </br></p> <p> Як небо кольору надій, </br> Що вже не буде більше хуги, </br> Як зір з-під пелюсткових вій </br> Моєї скромної подруги. </br> </body> </html> </pre>	<p>Я в ліс по проліски пішов. Це не далеко від Полтави. Що ж, проліски – моя любов. Ще рання. Хай і не яскрава.</p> <p>Цвітуть на пагорбках вони, Оточені кругом водою, Як перші вісники весни З їх радістю і їх журбою.</p> <p>Як небо кольору надій, Що вже не буде більше хуги, Як зір з-під пелюсткових вій Моєї скромної подруги.</p>
--	---

Тег, що забороняє перенос рядка, – `<NOBR>...</NOBR>`. Увесь текст, поміщений усередині цього тега, буде виведений браузером в один рядок. Можна все ж таки дозволити браузеру перевести рядок у визначеному місці тексту усередині цього тега, якщо поставити у відповідному місці тег `<WBR>`.

<pre> <html> <head> <title>Моя перша Web-сторінка</title> </head> <body> <p> <NOBR> Даний рядок </pre>	<p>Даний рядок є найдовшим рядком документа, який не допускає будь-якого розбиття, де б то не було.</p>
--	---


<pre> є найдовшим рядком документа, <WBR> який не допускає будь-якого розбиття, де б то не було. </NOBR></p> </body> </html> </pre>	
---	--

Теги заголовків `<h1>`–`<h6>` - теги контейнери. Текст, поміщений у них, виводиться як заголовок. Має параметр вирівнювання по горизонталі, аналогічний тегу `<p>`. Закриваюча частина обов'язкова.

<pre> <html> <head> <title> Моя перша Web-сторінка. Заголовки</title> </head> <body> <p><h1>заголовок першого рівня </h1></p> <p><h2>заголовок другого рівня </h2></p> <p><h3>заголовок третього рівня </h3></p> <p><h4>заголовок четвертого рівня </h4></p> <p><h5>заголовок п'ятого рівня </h5></p> </body> </html> </pre>	<p>заголовок першого рівня</p> <p>заголовок другого рівня</p> <p>заголовок третього рівня</p> <p>заголовок четвертого рівня</p> <p>заголовок п'ятого рівня</p>
--	--

Тег горизонтальної лінії `<hr>`. Рисує горизонтальну лінію і має параметри:

- `ALIGN` – вирівнювання із значеннями `LEFT`, `RIGHT` або `CENTER`.
- `WIDTH` – встановлює довжину лінії в пікселях чи відсотках від вікна браузера.
- `SIZE` – встановлює ширину лінії в пікселях.
- `NOSHADOW` – відмінняє рельєфність лінії.
- `COLOR` – указує колір лінії у форматі `RGB` або стандартному імені.

<pre> <html> <head> <title>Моя перша Web-сторінка</title> </head> <body> <HR ALIGN=RIGHT WIDTH=55% SIZE=20 COLOR=F74C80>. </body> </html> </pre>	
--	---

4. Теги для форматування шрифту

Для надання фрагменту тексту смислового значення дуже зручно використовувати засоби зміни зображення шрифту.

Для виділення тексту напівжирним шрифтом застосовується тег `` чи ``, для виділення курсивом – `<i>` або ``, для підкреслення тексту – `<u>`, для закреслення тексту – `<STRIKE>`. Фрагменти напівжирного або курсивного тексту використовують для введення нових термінів, вираження важливих думок. З підкресленням тексту треба бути дуже обережним, оскільки в Інтернеті таким чином прийнято виділяти гіперпосилання.

Тег `<CENTER>` дозволяє центрувати текст на сторінці по горизонталі. Це зручно для виділення найменувань чи заголовків.

HTML дозволяє одночасно застосовувати декілька тегів, призначених для зміни зображення шрифту. Для перетворення тексту в надрядковий індекс використовують тег `<SUP>`, для підрядкового індексу – `<SUB>`. Ці теги стануть в нагоді при введенні математичних текстів, хімічних формул, наукових статей, побудові виносок тощо. Зазвичай індекси відображаються шрифтом меншого розміру порівняно з основним. Цю обставину слід урахувати при виборі розміру основного шрифту, оскільки при малому розмірі індекси можуть бути просто не помітні.

<pre><html> <head> <title>Форматування</title> </head> <body> <p>Жирний текст</p> <p>Курсив</p> <p><u>Підкреслений текст</u></p> <p><strike>Закреслений текст</strike></p> <center>Центрування</center> <p>Жирний курсив</p> <p>Верхній <sup>індекс</sup></p> <p>Нижній <sub>індекс</sub></p> </body> </html></pre>	<p>Жирний текст</p> <p><i>Курсив</i></p> <p><u>Підкреслений текст</u></p> <p>Закреслений текст</p> <p>Центрування</p> <p><i>Жирний курсив</i></p> <p>Верхній ^{індекс}</p> <p>Нижній _{індекс}</p>
---	---

Для зміни шрифту служить тег `` з атрибутом FACE. Можна вказувати шрифт за назвою (Arial, Tahoma, Verdana) або типом (наприклад, monospace).

<pre><html> <head> <title> Моя перша Web-сторінка</title> </head> <body> <p>Форматування тексту</p> <p>Форматування тексту</p> <p>Форматування тексту</p> </body> </html></pre>	<p>Форматування тексту</p> <p><i>Форматування тексту</i></p> <p>Форматування тексту</p>
---	---

Для зміни розміру шрифту використовують атрибут **SIZE**. Чим більше шрифт, тим легшим для сприйняття стає текст. Проте дрібний шрифт дозволяє умістити в межах екрана більший об'єм інформації. Використовуючи даний атрибут із тегом **<BASEFONT>**, можна змінити розмір тексту Web-сторінки цілком. Застосовуючи атрибут **SIZE** спільно з тегом ****, можна впливати на зовнішній вигляд окремого фрагмента тексту. Доступні 7 розмірів шрифтів. Найменший позначається цифрою 1, а найбільший – 7.

Змінювати шрифт можна за допомогою тегів **<BIG>** і **<SMALL>**. **<BIG>** укрупнює шрифт указанного фрагмента по відношенню до попереднього тексту, а **<SMALL>** відповідно зменшує.

<pre><html> <head> <title> Моя деручи Web-сторінка</title> </head> <body> <p>Шрифт розміру 1</p> <p>Шрифт розміру 2</p> <p>Шрифт розміру 3</p> <p>Шрифт розміру 4</p> <p>Шрифт 5</p> <p>Шрифт 6</p> <p>Шрифт 7</p> </body> </html></pre>	<p>Шрифт розміру 1</p> <p>Шрифт розміру 2</p> <p>Шрифт розміру 3</p> <p>Шрифт розміру 4</p> <p>Шрифт 5</p> <p>Шрифт 6</p> <p>Шрифт 7</p>
--	--


Використовуючи атрибут **TEXT** в контексті тега **<BODY>**, можна змінити колір тексту Web-сторінки цілком. Застосування ж атрибуту **COLOR** з тегом **** (має вищий пріоритет) дозволяє впливати на зовнішній вигляд окремого фрагмента тексту.

Щоб змінити колір тексту, треба знати найменування кольору або його шістнадцятиричний код. Код складається із символу фунта # і числа.. По імені можна послатися тільки на 16 стандартних кольорів.

Найменування кольору	Код
aqua	##00FFFF
black	##000000
blue	##0000FF
fuchsia	##FF00FF
gray	##808080
green	##008000
lime	##00FF00
maroon	##800000
navy	##000080
olive	##808000
purple	##800080
red	##FF0000
silver	##C0C0C0
teal	##008080
white	##FFFFFF
yellow	##FFFF00

<pre><html> <head> <title>Моя перша Web-сторінка</title> </head> <body> <p>Червоний колір</p> <p>Зелений колір</p> <p>Синій колір </p> <p>Синій колір </p> </body> </html></pre>	<p>Червоний колір</p> <p>Зелений колір</p> <p>Синій колір</p> <p>Синій колір</p>
--	--

Атрибут `BGColor` дає можливість явної вказівки кольору фону Web-сторінки. Колір фону задається аналогічно кольору тексту (ім'ям або кодом).

<pre><html> <head> <title>Моя перша Web-сторінка</title> </head> <body BGColor="#00FF00"> <p> Червоний колір </p> <p> Зелений колір </p> <p> Синій колір </p> <p> Синій колір </p> </body> </html></pre>	 <p>Червоний колір</p> <p>Зелений колір</p> <p>Синій колір</p> <p>Синій колір</p>
--	---

5. Теги створення списків

У HTML передбачено три типи списків і відповідні цим типам дескриптори:

- Маркований (невпорядкований) список;
- Нумерований (упорядкований) список;
- Список визначень.

Маркований (невпорядкований) список

Неврегульований список є списком елементів. Елементи списку маркуються за допомогою спеціальних знаків (зазвичай невеликий чорний круг).

Неврегульований список починається з тега ``. Кожний елемент списку починається з тега ``.

<pre><html> <body> <h4>Неврегульований список:</h4> елемент 1 елемент 2 елемент 3 </body> </html></pre>	Неврегульований список: <ul style="list-style-type: none">• елемент 1• елемент 2• елемент 3
---	--

За умовчанням елементи списку маркуються чорним кружечком. За допомогою атрибуту `TYPE` можна змінити стиль маркування.

<pre><HTML> <HEAD> <BODY> <ul type="circle"> елемент 1 елемент 2 <li type="disc">елемент 3 <li type="square">елемент 4 </BODY></HTML></pre>	<ul style="list-style-type: none">○ елемент 1○ елемент 2• елемент 3▪ елемент 4
---	---

Нумерований (упорядкований) список

Упорядкований список також є списком елементів. Елементи списку маркуються за допомогою чисел або букв.

Упорядкований список починається з тега ``. Кожний елемент списку починається з тега ``. У тега `` може бути два атрибути `start` (визначає перше число, з якого починається нумерація пунктів) і `type` (визначає стиль нумерації пунктів). Атрибут `type` може мати значення:

- "A" – великі букви A, B, C ...
- "a" – рядкові букви a, b, з ...
- "I" – великі римські числа I, II, III ...
- "i" – маленькі римські числа i, ii, iii ...
- "1" – арабські числа 1, 2, 3 ...

<pre> <html> <body> <h4>Список із цифрами:</h4> принтер Canon принтер Epson принтер Samsung <h4>Список із буквами:</h4> <ol type="B"> принтер Canon принтер Epson принтер Samsung <h4>Список із буквами нижнього регiстра:</h4> <ol type="a"> принтер Canon принтер Epson принтер Samsung <h4>Список із римськими цифрами:</h4> <ol type="I"> принтер Canon принтер Epson принтер Samsung <h4>Список із римськими цифрами в нижньому регiстрі:</h4> <ol type="i"> принтер Canon принтер Epson принтер Samsung </body> </html> </pre>	<p>Список із цифрами:</p> <ol style="list-style-type: none"> 1. принтер Canon 2. принтер Epson 3. принтер Samsung <p>Список із буквами:</p> <ol style="list-style-type: none"> B. принтер Canon C. принтер Epson D. принтер Samsung <p>Список із буквами нижнього регiстра:</p> <ol style="list-style-type: none"> a. принтер Canon b. принтер Epson c. принтер Samsung <p>Список із римськими цифрами:</p> <ol style="list-style-type: none"> I. принтер Canon II. принтер Epson III. принтер Samsung <p>Список з римськими цифрами в нижньому регiстрі:</p> <ol style="list-style-type: none"> i. принтер Canon ii. принтер Epson iii. принтер Samsung
--	--

Багаторівневий список

Для створення багаторівневих списків можна використовувати як марковані, так і нумеровані списки, а також їх поєднання. Багаторівневий список формується шляхом вкладення одного списку в тіло іншого.

Списки визначень

Списки визначень створюються за допомогою тега-контейнера `<DL>...</DL>`. У середині контейнера визначуваний термін позначається тегом `<DT>`, а абзац із його визначенням – тегом `<DD>`. Для тегів `DT` і `DD` закриваючі елементи не обов'язкові. Приклад такого списку:

<pre><html> <body> <h4>Список визначень:</h4> <dl> <dt>елемент 1</dt> <dd>опис елемента 1</dd> <dt>елемент 2</dt> <dd>опис елемента 2</dd> </dl> </body></pre>	<p>Список визначень: елемент 1 опис елемента 1 елемент 2 опис елемента 2</p>
--	---

6. Таблиці HTML

Опис таблиць повинен розташовуватися всередині розділу документа `<BODY>`. Документ може містити довільне кількість таблиць, причому допускається і вкладеність таблиць один в одному. Кожна таблиця має починатися тегом `<TABLE>` і завершуватися тегом `</TABLE>`. У середині цієї пари тегів розташовується опис вмісту таблиці. Будь-яка таблиця складається з одного або декількох рядків, у кожному з яких задаються дані для окремих клітинок.

Для тега `table` визначені такі атрибути:

- `align` – визначає спосіб горизонтального вирівнювання таблиці на сторінці. Можливі значення: `left`, `center`, `right`. Значення за умовчанням – `left`;
- `valign` – повинен установлювати спосіб вертикального вирівнювання для вмісту таблиці. Можливі значення: `top`, `bottom`, `middle`;
- `border` – визначає ширину зовнішньої рамки таблиці (у пікселях). При `BORDER="0"` чи за відсутності цього параметра рамка відобразиться не буде;
- `cellpadding` – установлює відстань (у пікселях) між рамкою кожного елемента таблиці та матеріалом, що міститься в ній;
- `cellspacing` – визначає відстань (у пікселях) між межами сусідніх клітинок;

- `width` – установлює ширину таблиці. Ширина задається або в пікселях, або в процентному відношенні до ширини вікна браузера. За умовчанням цей параметр визначається автоматично залежно від об'єму матеріалу, що міститься в таблиці;
- `height` – установлює висоту таблиці. Висота задається або в пікселях, або в процентному відношенні до висоти вікна браузера. За умовчанням цей параметр визначається автоматично залежно від об'єму матеріалу, який міститься в таблиці;
- `bgcolor` – установлює колір фону елементів таблиці. Задається або RGB-значенням у шістнадцятковій системі, або одним з 16 базових кольорів.
- `background` – дозволяє заповнити фон таблиці малюнком. Як значення необхідно вказати URL малюнка.

Кожний рядок починається тегом `<TR>` (Table Row) і завершується тегом `</TR>`. Окрема комірка у рядку обрамляється парою тегів `<TD>` та `</TD>` (Table Data) чи `<TH>` і `</TH>` (Table Header). Тег `<TH>` використовується зазвичай для елементів-заголовків таблиці, а `<TD>` – для комірок-даних.

<pre> <html> <body> <h1>Це приклад простої таблиці, що містить один рядок і одну комірку.</h1> <table border="1"> <tr> <td>Один рядок та одна комірка</td> </tr> </table> <h1>Один рядок і три стовпці:</h1> <table border="1"> <tr> <td> стовпець 1</td> <td> стовпець 2</td> <td> стовпець 3</td> </tr> </table> <h1>Два рядки та три стовпці:</h1> <table border="1"> <tr> <td>1.1</td> <td>1.2</td> <td>1.3</td> </tr> <tr> <td>2.1</td> <td>2.2</td> <td>2.3</td> </tr> </table> </body> </html> <h1>Рамка таблиці</h1> <html> <body> <h1>Звичайна рамка:</h1> </pre>	<p>Це приклад простої таблиці, що містить один рядок і один осередок.</p> <table border="1"> <tr> <td>Один рядок та одну комірку</td> </tr> </table> <p>Один рядок і три стовпці:</p> <table border="1"> <tr> <td>стовпець 1</td> <td>стовпець 2</td> <td>стовпець 3</td> </tr> </table> <p>Два рядки та три стовпці:</p> <table border="1"> <tr> <td>1.1</td> <td>1.2</td> <td>1.3</td> </tr> <tr> <td>2.1</td> <td>2.2</td> <td>2.3</td> </tr> </table> <p>Рамка таблиці</p> <p>Звичайна рамка:</p> <table border="1"> <tr> <td>Перший</td> <td>рядок</td> </tr> <tr> <td>Другий</td> <td>рядок</td> </tr> </table> <p>Товста рамка:</p> <table border="2"> <tr> <td>Перший</td> <td>рядок</td> </tr> <tr> <td>Другий</td> <td>рядок</td> </tr> </table>	Один рядок та одну комірку	стовпець 1	стовпець 2	стовпець 3	1.1	1.2	1.3	2.1	2.2	2.3	Перший	рядок	Другий	рядок	Перший	рядок	Другий	рядок
Один рядок та одну комірку																			
стовпець 1	стовпець 2	стовпець 3																	
1.1	1.2	1.3																	
2.1	2.2	2.3																	
Перший	рядок																		
Другий	рядок																		
Перший	рядок																		
Другий	рядок																		

```

<table border="1">
<tr>
  <td>Перший</td>
  <td>рядок</td>
</tr>
<tr>
  <td>Другий </td>
  <td>рядок</td>
</tr>
</table>
<h1>Товста рамка:</h1>
<table border="10">
<tr>
  <td>Перший</td>
  <td>рядок</td>
</tr>
<tr>
  <td>Другий </td>
  <td>рядок</td>
</tr>
</table>

</body>
</html>

```

Для створення заголовка таблиці служить тег `<CAPTION>`.

За умовчанням браузер розташовує заголовок таблиці по центру над нею. За допомогою атрибута `ALIGN` із значенням `bottom` можна розмістити заголовок під таблицею.

```

<html>
<body>
<table border="1">
<caption> Заголовок таблиці </caption>
<tr>
  <td>1</td>
  <td>2</td>
</tr>
</table>
</body>
</html>

```

Злиття елементів таблиці

За допомогою атрибутів `COLSPAN`, `ROWSPAN` можна об'єднувати декілька сусідніх комірок стовпця або рядка в одну велику комірку.

Атрибут `COLSPAN` тегів `<TH>`, `<TD>` дозволяє зробити злиття комірок декількох стовпців у межах одного рядка.

Атрибут `ROWSPAN` – клітинок із декількох рядків у межах одного стовпця.

Цифра, що зазначається в цих атрибутах, указує кількість комірок, які будуть піддані злиттю.

Оскільки при виконанні операцій злиття дуже легко заплутатися, то настійно рекомендується зробити попередній ескіз таблиці на папері.

<pre> <html> <body> <table border="1"> <tr> <td rowspan="3">1111</td> <td colspan="2">22222</td> <td colspan="2">33333</td> </tr> <tr> <td>44444</td> <td colspan="2" rowspan="2">55555</td> <td>66666</td> </tr> <tr> <td>77777</td> <td>88888</td> </tr> <tr> <td colspan="5">99999</td> </tr> </table> </body> </html> </pre>	<table border="1"> <tr> <td></td> <td>22222</td> <td>33333</td> <td></td> <td></td> </tr> <tr> <td>1111</td> <td>44444</td> <td rowspan="2">55555</td> <td>66666</td> <td></td> </tr> <tr> <td></td> <td>77777</td> <td>88888</td> <td></td> </tr> <tr> <td colspan="5">99999</td> </tr> </table>		22222	33333			1111	44444	55555	66666			77777	88888		99999				
	22222	33333																		
1111	44444	55555	66666																	
	77777		88888																	
99999																				

7. Гіперпосилання

Для створення гіперпосилання служить тег `<a>`, який називають «якір» (anchor). Цей тег має два обов'язкові атрибути. Для того щоб створити посилання на заданий елемент усередині WEB-сторінки на іншу WEB-сторінку, Internet-ресурс і т.п., тег `<a>` потрібно використовувати з атрибутом `href`:

``

Якщо потрібно зробити закладку (якір), на котру потім буде зроблено посилання, тег `<a>` необхідно використовувати з атрибутом `name`:

`<a name ="елемент документа, на який потрібно зробити посилання"`

Для переходу на місце, відмічене іменованим якорем використовується знак `#`.

<pre> <html> <head> <title>Форматування</title> </head> <body> <p>Список розділів</p> Розділ 1 Розділ 2 <p>Розділ 1</p> <p>Текст розділу 1</p> <p>Розділ 2</p> <p>Текст розділу 2
</p> </body> </html> </pre>	<p>Список розділів</p> <ul style="list-style-type: none"> • Розділ 1 • Розділ 2 <p>Розділ 1</p> <p style="padding-left: 40px;">Текст розділу 1</p> <p>Розділ 2</p> <p style="padding-left: 40px;">Текст розділу 2</p>
--	--

Посилання можна зробити і на будь-яку електронну адресу. Для цього потрібно в код посилання включити префікс `mailto`.

<pre><html> <body> <p> Це посилання на поштову адресу: відправити запит </p> <p>За допомогою цього посилання відправляється лист за адресою електронної пошти tnd@mail.ru </p> </body> </html></pre>	<p>Це посилання на поштову адресу: отправити запит</p> <p>За допомогою цього посилання відправляється лист за адресою електронної пошти tnd@mail.ru</p>
--	--

За допомогою атрибута `TARGET` можна завантажити сторінку в новому вікні браузера. Цей атрибут призначений для задання найменування вікна. Ім'я вікна використовується в службових цілях. Щоб відкрити сторінку в новому вікні, треба застосовувати константу `_blank`.

<pre><html> <body> Остання сторінка <p> Якщо задати атрибут target посилання як "_blank", то посилання буде відкрито в новому вікні. </p> </body> </html></pre>	<p>Остання сторінка</p> <p>Якщо задати атрибут target посилання як "_blank", то посилання буде відкрито в новому вікні.</p>
--	---

8. Робота з графікою

Для того щоб уставити у Web-сторінку зображення, необхідно або намалювати його, або взяти вже готове. Графіка повинна бути представлена у вигляді файлів із розширеннями `.jpg` `.gif`.

Для вставки графічних елементів служить тег `` з обов'язковим атрибутом `SRC` і декількома необов'язковими. Тег `SRC` виконує важливу роль у графічному зображенні на сторінці — він задає шлях (відносний чи абсолютний) до малюнка.

Необов'язкові параметри:

- Параметр `ALT=""` як значення містить напис, що розповідає про зміст зображення для тих відвідувачів, браузери яких не підтримують графіку або працюють у режимі відключеної графіки. Цей же напис з'являється при наведенні покажчика мишки на зображення.

- Параметри `WIDTH=""` `HEIGHT=""` як значення мають розміри зображення по ширині й висоті в пікселях. Ці параметри бажано вказувати в кодї сторінки, щоб браузер заздалегідь залишав місце для зображення, тоді, при завантаженні сторінка менше "сіпатиметься". Крім того, ці параметри можна вживати для регулювання розмірів малюнка у вікні браузера.
- Параметр `BORDER=""` промальовував рамку навколо зображення. Як значення вказується цифра, що позначає ширину рамки в пікселях.
- Параметр `ALIGN=` визначає положення зображення на сторінці. Про значення цього параметра необхідно поговорити докладніше.

HTML дозволяє легко додати на сторінку фонове зображення за допомогою атрибута `BACKGROUND` тега `<BODY>`.

<pre> <html> <body> <p>Варіанти вирівнювання</p> <p> Вирівнювання по верхній кромці</p> <p> Вирівнювання посередині</p> <p> Вирівнювання по нижній кромці</p> <p>Варіанти розміщення тексту навколо малюнка</p> Текст розміщується праворуч

 Текст розміщується ліворуч

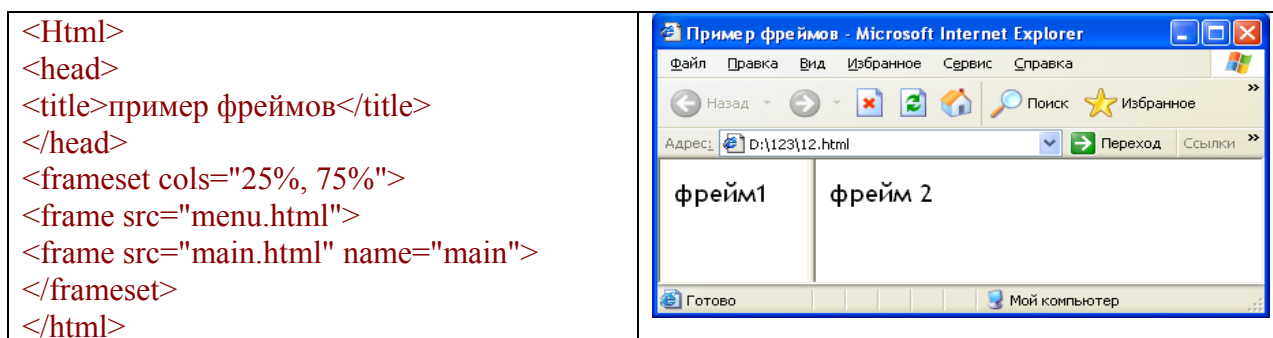
 Текст між малюнками

 <p>Варіанти зміни розміра малюнка</p>
 </body> </html> </pre>	<p>Варіанти вирівнювання</p> <p> Вирівнювання по верхній кромці</p> <p> Вирівнювання посередині</p> <p> Вирівнювання по нижній кромці</p> <p>Варіанти розміщення тексту навколо малюнка</p> <p> Текст розміщується праворуч</p> <p> Текст розміщується ліворуч</p> <p> Текст між малюнками</p> <p>Варіанти зміни розміра малюнка</p> <p> </p>
---	---

9. Фрейми

Фрейми – це засіб форматування сторінки. Вони дозволяють розділити вікно перегляду броузера на декілька прямокутних областей, у яких відображається різна інформація. Фрейми визначаються в структурі `<FRAMESET>`, котра використовується для сторінок, що містять фрейми, замість розділу `<BODY>` звичайного документа. Web-сторінки, складені з фреймів, не можуть містити розділ `<BODY>` в HTML-кодi. У свою чергу, сторінки з розділом `<BODY>` не можуть використовувати фрейми. Загальний контейнер `FRAMESET` описує всі фрейми, на які ділиться екран. У середині тега-контейнера `<FRAMESET>...</FRAMESET>` можуть міститися тільки вкладені теги `FRAMESET` і теги `FRAME`. Тег `FRAMESET` має два взаємовиключальні параметри: `ROWS` та `COLS`.

Горизонтальне ділення екрана задається за допомогою атрибута `rows`, а вертикальне – за допомогою атрибута `cols`. Значення атрибута можуть бути виражені в пікселях або відсотках. Крім того застосовується символ `*` для позначення частини екрана, що залишилася.



Тег `FRAME` описує кожний фрейм окремо. Тег `<FRAME>` має шість параметрів: `src`, `name`, `marginwidth`, `marginheight`, `scrolling` і `noresize`.

Значення параметра `src` визначає URL-адреса документа, яка буде завантажена спочатку в даний фрейм. Параметр `name` встановлює ім'я фрейма, котре може використовуватися для посилання до даного фрейма. Зазвичай посилання задається з іншого фрейма, розташованого на тій же самій сторінці.

Для заборони зміни меж фрейма застосовується атрибут `noresize`. Це зручно, якщо треба заборонити користувачеві модифікувати початкове компонування сторінки. За умовчанням користувач може переміщати межі фрейма за своїм бажанням, задаючи нові розміри об'єкта, щоб краще розглянути певний фрагмент даних.

Атрибут `marginwidth` дозволяє задавати ширину лівого і правого поля.

Атрибут `marginheight` – висоту верхнього та нижнього поля.

Задаючи величини полів, треба використовувати обидва атрибути, оскільки броузер може автоматично скоротити розміри тих полів, які ви не вказали, до найменшого значення.

За умовчанням браузер відображає смуги прокрутки тільки в тому випадку, якщо дані не укладаються в розміри фрейма. Надавши атрибуту `scrolling` значення `yes/no`, можна наказати браузеру завжди зберігати смуги прокрутки в прихованому стані або постійно відображати їх.

10.Форми

Форми HTML призначені для організації взаємодії з користувачем. Вони дозволяють вводити текст, здійснювати вибір із запропонованих значень за допомогою списків чи кнопок. За допомогою форм можна організувати інтерактивний обмін інформацією між Web-сторінкою і сервером. Можна визначити форми як електронні бланки для заповнення різних даних, таких як, наприклад, ім'я, вік, вибір країни мешкання й ін. Як правило, форма працює спільно зі встановленим на сервері сценарним застосуванням, що обробляє введену інформацію.

Форма є областю, яка може містити елементи, що дозволяють користувачеві вводити інформацію (такі, як текстові поля, поля багаторядкового тексту, меню, які розкриваються, перемикачі, прапорці тощо).

Форма визначається за допомогою тегів `<form></form>`, між котрими розташовуються поля введення, кнопки, а також усі необхідні елементи оформлення форми.

Тег `<form>` має ряд атрибутів, з яких необхідно виділити атрибути `action` і `method`. Без цих атрибутів форма не зможе передати інформацію від користувача на сервер.

```
<form action="html_form_action.asp" method=get>
```

Атрибут `Action` указує URL-адресу об'єкта, котрий повинен отримати дані форми.

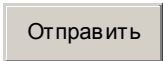
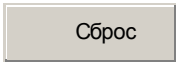
Атрибут `method` може мати два значення: `get` і `post`.

Значення атрибута `method=get` примушує Web-браузер передати всі дані формуляру за URL-адресою, заданою в `action`. При цьому введені при заповненні форми дані просто додаються в адресний рядок із використанням роздільника – знака питання. Цей метод зручний для невеликих форм.

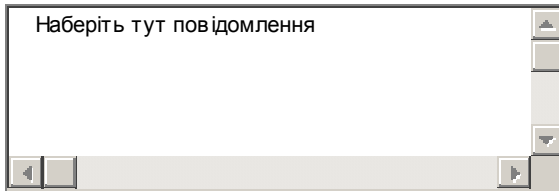
Значення атрибута `method=post` примушує Web-браузер, перш за все, зв'язатися із сервером, що оброблює форму, і лише після встановлення зв'язку приступити до передачі даних, для обробки яких використовуватимуться спеціальні сценарії.

Для внесення інформації користувачем до форми застосовується елемент `<INPUT>`. Це і є поля, в які користувач вводить інформацію. Кожний елемент `<INPUT>` включає атрибут `NAME=ім'я`, що визначає ім'я даного поля (ідентифікатор поля). У таблиці представлені основні типи вживаних елементів `<INPUT>`:

<p>TYPE=text</p>	<p>Визначає вікно для введення рядка тексту. Може містити додаткові атрибути <code>SIZE=число</code> (ширина вікна введення в символах) і <code>MAXLENGTH=число</code> (максимально допустима довжина рядка, що вводитьься, в символах):</p> <pre><INPUT TYPE=text SIZE=20 NAME=User VALUE="LENIN INC"></pre> <div data-bbox="450 501 611 546" style="border: 1px solid black; padding: 2px; width: fit-content;">LENIN INC</div> <p>Визначає вікно шириною 20 символів для введення тексту. За умовчанням у вікні знаходиться текст LENIN INC, який користувач може змінити.</p>
<p>TYPE=password</p>	<p>Визначає вікно для введення пароля. Абсолютно аналогічний типу text, тільки замість символів тексту, що вводитьься, показує на екрані зірочки (*):</p> <pre><INPUT TYPE=password NAME=PW SIZE=20 MAXLENGTH=10></pre> <div data-bbox="450 936 611 981" style="border: 1px solid black; width: 100px; height: 20px;"></div> <p>Визначає вікно шириною 20 символів для введення пароля. Максимально допустима довжина пароля — 10 символів.</p>
<p>TYPE=radio</p>	<p>Визначає радіокнопку. Може містити додатковий атрибут CHECKED (показує, що кнопка відмічена). У групі радіокнопок з однаковими іменами може бути тільки одна помічена радіокнопка:</p> <pre><INPUT TYPE=radio NAME=Question VALUE="Yes" CHECKED> Так <INPUT TYPE=radio NAME=Question VALUE="No"> Ні <INPUT TYPE=radio NAME=Question VALUE="Possible"> Можливо</pre> <div data-bbox="450 1485 636 1637"> <input checked="" type="radio"/> Так <input type="radio"/> Ні <input type="radio"/> Можливо </div> <p>Визначає групу з трьох радіокнопок, підписаних Yes, No і Possible. Спочатку помічена перша з кнопок. Якщо користувач не відзначить іншу кнопку, обробникові буде передана змінна Question із значенням Yes. Якщо користувач відзначить іншу кнопку, обробникові буде передана змінна Question із значенням No або Possible.</p>
<p>TYPE=checkbox</p>	<p>Визначає квадрат, в якому можна зробити позначку. Може містити додатковий атрибут CHECKED (показує, що квадрат помічений). На відміну від радіокнопок, в групі квадратів з однаковими іменами може бути декілька помічених квадратів:</p>

	<pre><INPUT TYPE=checkbox NAME=Comp VALUE="CPU"> Процесори <INPUT TYPE=checkbox NAME=Comp VALUE="Video" CHECKED> Відеоадаптери <INPUT TYPE=checkbox NAME=Comp VALUE="Scan"> Сканери <INPUT TYPE=checkbox NAME=Comp VALUE="Modem" CHECKED> Модеми</pre> <p><input type="checkbox"/> Процесори <input checked="" type="checkbox"/> Відеоадаптери <input type="checkbox"/> Сканери <input checked="" type="checkbox"/> Модеми</p> <p>Визначає групу з чотирьох квадратів. Спочатку помічені другий і четвертий квадрати. Якщо користувач не проведе змін, обробникові будуть передані дві змінні: Comp=Video і Comp=Modem.</p>
TYPE=hidden	<p>Визначає прихований елемент даних, який не видно користувачеві при заповненні форми і передається обробникові без змін. Такий елемент іноді корисно мати у формі, котра час від часу піддається переробці, щоб обробник міг знати, з якою версією форми він має справу.</p> <pre><INPUT TYPE=hidden NAME=version VALUE="1.1"></pre> <p>Визначає приховану змінну version, що передається обробникові із значенням 1.1.</p>
TYPE=submit	<p>Визначає кнопку, при натисненні на яку запускається процес передачі даних із форми обробникові:</p> <pre><INPUT TYPE=submit VALUE="Отправить"></pre> <p></p>
TYPE=reset	<p>Визначає кнопку, при натисненні на котру очищаються поля форми. Оскільки при використанні цієї кнопки дані обробникові не передаються, кнопка типу reset може і не мати атрибута NAME:</p> <pre><INPUT TYPE=reset VALUE=" Скидання "></pre> <p></p>

Форми можуть містити поля для введення великого тексту <TEXTAREA>:
 <TEXTAREA NAME=address ROWS=5 COLS=50> Наберіть тут повідомлення</TEXTAREA>



Атрибут `NAME` визначає ім'я, під яким уміст вікна буде переданий обробникові. Атрибут `ROWS` встановлює висоту вікна в рядках. Атрибут `COLS` визначає ширину вікна в символах. Текст, розміщений між тегами `<TEXTAREA></TEXTAREA>`, є вмістом вікна за умовчанням. Користувач може його відредагувати або просто стерти.

Крім усього цього, форми можуть містити меню вибору, яке починається тегом `<SELECT>`, що відкривається (містить обов'язковий атрибут `NAME`, який визначає ім'я меню) і завершується `</SELECT>`, що закривається. Між ними знаходяться теги `<OPTION>`, котрі визначають елемент меню. Обов'язковий атрибут `VALUE` встановлює значення, яке буде передано обробникові, якщо вибраний цей елемент меню. Тег `<OPTION>` може включати атрибут `selected`, котрий показує, що даний елемент позначений за умовчанням.

<pre><SELECT NAME="имя"> <OPTION VALUE="option_1" selected>текст 1 <OPTION VALUE="option_2">текст 2 <OPTION VALUE="option_n">текст n </SELECT></pre>	
--	--

Тег `<SELECT>` може також містити атрибут `MULTIPLE`, наявність якого показує, що з меню можна вибрати декілька елементів. Більшість оглядачів показують меню `<SELECT MULTIPLE>` у вигляді вікна, в котрому знаходяться елементи меню. Висоту вікна в рядках можна задати атрибутом `SIZE=число`.

<pre><SELECT MULTIPLE SIZE=3 NAME="им'я"> <OPTION VALUE="option_1" selected>текст 1 <OPTION VALUE="option_2">текст 2 <OPTION VALUE="option_n">текст n </SELECT></pre>	
--	--

Завдання для лабораторних занять

Завдання 1. Простий документ HTML

За допомогою Блокнота Windows створіть і збережіть в особистій папці HTML-файл з ім'ям index.html та заголовком, що збігається з Вашим прізвищем. У тексті файлу створіть 2 заголовки першого рівня "Мої анкетні дані" і "Мое хобі".

У розділах "Мої анкетні дані" та "Мое хобі" додайте по 2–3 абзаци тексту, вирівняних за допомогою опції align=left | right | center | justify. У середині тексту використовуйте виділення слів курсивом, жирним і підкресленням.

До нижньої частини документа додайте рядок, що не розривається, набраний шрифтом Arial ыз використанням розміру 4. Додайте до документа підпис, відформатований таким чином:

Студент(ка) групи N Іванова І.І.

Перегляньте отриманий файл за допомогою броузера Internet Explorer. Зверніть увагу на зміну форматування документа при змінах розміру вікна і розміру шрифтів браузера.

Завдання 2. Документ HTML із посиланнями

Створіть документ links.html із заголовком "Мої посилання" та збережіть його в тій же папці, що і документ index.html із завдання 1. Додайте в links.html перелік посилань на різні ресурси Інтернет, наприклад:

Поштова служба www.mail.ru

Пошукові сервери [Яndex](#) і [Rambler](#)

і так далі – всього 6–8 посилань. При клацанні по них посилання повинні відкриватися в новому вікні.

Додайте у верхню частину документа links.html посилання вигляду **[на головну сторінку](#)**

що повертає до документа index.html

Додайте в документ index.html рядок виду "", що адресує документ links.html. Документ links.html повинен відкриватися в поточному вікні.

Зробіть підпис із файлу index.html посиланням на Вашу адресу електронної пошти.

Додайте в нижню частину файлу index.html посилання "в початок сторінки", що адресує перший рядок документа.

Завдання 3. Створення списків

Зробіть перелік посилань із документа links.html маркованим списком із маркерами у вигляді квадрата.

Додайте до документа links.html список з 4–5 визначень, що складається з посилань і коментарів до них. Приклад елемента такого списку:

[Яndex](#)

[Яndex](#) – це найпопулярніший в Росії пошуковий сервер

Завдання 4. Створення таблиць

Створіть документ `tab.html`. Основна частина документа повинна бути організована як таблиця, що складається з 2-х колонок. У лівій колонці розташовуються посилання, призначені для зв'язку документа з іншими сторінками сайту, а в правій колонці – основний зміст. Ширину лівої колонки виберіть в межах 150 – 200 пікселів, ширину всієї таблиці – 90 – 100% від ширини вікна броузера, обрамлення таблиці повинне бути відключене. Наприклад, таблиця може виглядати так:

[На головную...](#)

Основна частина документа знаходиться в правій колонці!

Завдання 5. Робота з кольором і графікою

Змініть основний колірний набір документа `links.html`. Частини документа оформіть різними кольорами.

Створіть за допомогою редактора Paint малюнок, конвертуйте його з формату `bmp` і розмістіть на Вашій сторінці.

За допомогою таблиць і фонових зображень оформіть свої сторінки.

Завдання 6. Робота з формами

Створіть документ HTML, що містить форму для заповнення замовлення на товар. Замовлення повинне бути відправлене електронною поштою.

Форма передбачає поле введення номера банківського рахунку, поле введення особистого номера користувача (цей номер слід захистити від "підглядання" при введенні), список вибору товарів (можна вибрати як один, так і декілька товарів), поле введення довільного коментаря користувача.

Також користувач має вказати один із двох методів оплати – готівковим або безготівковим розрахунком. Користувач повинен мати змогу вказати, яку додаткову інформацію він хоче отримати при відповіді – дані про залишки на рахунку і/або список останніх платежів по рахунку. Будь-яка із цих двох опцій може бути включена чи вимкнена.

Передбачте також можливість скидання введених у форму даних.

Завдання 7. Робота з фреймами

Створіть документ, що містить 2 вертикально розташованих фрейми. Лівий фрейм (вузький) призначений для виведення меню, правий (широкий) – для виведення вмісту документів, на які Ви посилаєтеся з меню. Розміри фреймів повинні бути змінні. Залиште невеликий вільний простір між фреймами.

Створіть у лівому фреймі меню для навігації по всіх сторінках Вашого сайту. Перевірте роботу документа з фреймами.

Контрольне завдання

1. Створіть набір HTML-сторінок, пов'язаних однорівневим меню (сторінкам дати назви "Головна сторінка", "Мої посилання" і "Програми"). Меню розташувати в рядок по центру екрана або в окремому стовпці таблиці.

2. Створіть сторінку, що містить три фрейми:

Ф рейм 1	Фрейм 2
	Фрейм 3

У фрейм 1 помістити гіперпосилання на два різні існуючі файли HTML. У фреймі 2 повинен розміщуватися заголовок розділу, а у фреймі 3 – текст.

3. Створіть сторінку із списком студентів (наприклад, список учасників якого-небудь студентського змагання). Для кожного студента вкажіть П.І.Б., місто, ВНЗ, факультет, курс, вік, адреса поштова, адреса електронної пошти. Зробіть так, щоб перелічені значення відрізнялися зовні один від одного, тобто, щоб, наприклад, всі назви ВНЗ відображались однотипно і відмінно від тих, що йдуть перед ними. Використовуйте для цього каскадні таблиці стилів.

4. Створіть стиль для оформлення сторінок, в який помістіть такі властивості: колір заголовка сторінки, шрифт і колір основного тексту, колір посилань непроглянутих, колір проглянутих посилань, колір активного посилання, фон сторінки. Застосуйте цей стиль до реальної HTML-сторінки.

5. У документі HTML створіть таблицю за вказаною схемою.

Оформіть фоновим кольором комірки першого рядка.

6. У документі HTML створіть таблицю за вказаною схемою.

Задайте ширину першого стовпця 33%.

7. Уставіть у новий HTML-документ такі об'єкти:

- П.І.Б. студента, група;
- фоновий малюнок (файл фону);
- таблицю на 2 стовпці, 3 рядки, створіть в кожній комірці свій фон і вставіть текст;
- малюнок у форматі GIF;
- червону лінію довжиною 600 пк, шириною 2 пк ;
- 2 радіокнопки;
- гіперпосилання на адресу електронної пошти;

8. Створіть головну сторінку сайта за зразком:



Торговий дім «Полтавський»

Пропонуємо широкий вибір товарів вітчизняного виробника

Звертатися за адресою:

м. Полтава, майдан Торгівлі, 1.

З нашими товарами можна ознайомитися на сторінці: [магазин](#)

Запрошуємо відвідати ресторан [«Полтавська кухня»](#)

9. Створіть таблицю, що демонструє коди кольорів (коди вписані в комірках і фон комірки забарвлений у той же колір).

	колір		колір
	FFFFFF0		FFFF80
	FFFFE0		FFFF60
	FFFFD0	0	FFFF40
	FFFC0	1	FFFF20
	FFFFB0	2	FFFF00
	FFFA0	3	FFFD00
	FFFF90	4	FFFC00

10. Створіть рекламну сторінку книжкового магазину.

11. Створіть сторінку-меню ресторану за зразком:

Меню ресторану «Полтавська кухня»

1. **Сніданок**
 - Макарони по-флотські.
 - Гречаники.
 - Рисова молочна каша.
2. **Обід**
 - 2.1. Перші страви:
 - суп харчо;
 - борщ;
 - суп гречаний;
 - суп із галушками.
 - 2.2. Другі страви:
 - котлета по-київськи;
 - шніцель по-полтавськи.
3. **Вечеря**
 - Кура-гриль.
 - Плов.
 - Шашлик.
4. **Напої**
 - Чай.
 - Кава.
 - Компот.
 - Сік.

12. Створіть форму-анкету, яка містить такі поля: Прізвище

Ім'я

Побатькові

E-mail

Вибір країни (обов'язково випадаючим **SELECT**, країн не менше 10-ти)

Вибір міста (обов'язково за допомогою **radio**, змінні повинні бути однаковими, не менше 5-ти)

Вибір мови (обов'язково за допомогою **checkbox**, змінні мають бути різноманітними, не менше 5-ти)

Вибір професій (обов'язково за допомогою **SELECT MULTIPLE**, змінні повинні бути різноманітними, не менше 10-ти)

Пароль

Додаткова інформація (обов'язково за допомогою **TEXTAREA**)

У прихованому полі (**hidden**), передайте змінну student із значенням "Ваше_ім'я"(student=Ваше_ім'я).

Кнопка для завантаження інформації на сервер

Кнопка для очищення форми

Дані з форми повинні посилатися обробникові електронною поштою.

Література

1. Матросов А.В., Сергеев А.О., Чаунин М.П. HTML 4.0. – СПб.: БХВ-Петербург, 2003. – 672 с.: ил.
2. Петюшкин А.В. HTML. Экспресс-курс. – СПб.: БХВ-Петербург, 2003. – 256 с.: ил.
3. <http://www.ngasu.nsk.su/prikl/html.html>
4. <http://w-wb.com/index.php>