

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ**

**К.О. Холупняк
Н.В. Ковальська**

**ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
HTML ДЛЯ ПОЧАТКІВЦІВ**

Черкаси – 2019

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

Основи інформаційних технологій. HTML для початківця.

УДК 004.43 (076)

*Рекомендовано до друку рішенням методичної ради
Черкаського державного бізнес-коледжу
Протокол № 3 від грудня 2019 р.*

Укладачі: Холупняк К.О., Ковальська Н.В.
Основи інформаційних технологій.
HTML для початківців.

Черкаси, 2019 р. – 64 с.

Рецензент: Андрієнко В.О. к.т.н., доцент кафедри комп'ютерних наук та системного аналізу Черкаського державного технологічного університету.

Навчальне видання розроблено у відповідності до навчальної програми «Інформатика» для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту), містить короткі теоретичні відомості та приклади завдань з теми «Web-програмування», призначений для використання при вивченні курсу «Основи інформаційних технологій». Навчальне видання містить завдання з таких розділів: створення web-сторінок, форматування тексту, створення списків, вставка графічних об'єктів, створення таблиць, гіперпосилань, фреймів. Посібник розрахований для початківців, людей, які тільки вирішили освоїти web-ремесло, може використовуватися студентами комп'ютерних спеціальностей.

Призначений для студентів закладів вищої освіти I-II рівнів акредитації.

Затверджено на засіданні
циклової комісії
комп'ютерних систем та мереж
Протокол № 4 від 22 листопада 2019 року

© К. О. Холупняк,
Н. В. Ковальська, 2019

ЗМІСТ

Вступ	4
Розділ 1. Створення web-сторінки. Форматування тексту, створення списків	5
Розділ 2. Вставка графічних об'єктів	21
Розділ 3. Створення таблиць	25
Розділ 4. Гіперпосилання	37
Розділ 5. Фрейми	46
Рекомендована література	62

ВСТУП

В навчальному виданні «Основи інформаційних технологій. HTML для початківців» розглядається мова гіпертекстової розмітки HTML, яка є основою при вивченні методів створення web-сторінки.

У навчальному виданні на доступній мові з множиною прикладів описано, як створити web-сторінку та потім повноцінний сайт, вставити в неї графіку, таблиці, використати посилання.

Розглядаються 5 розділів, в яких описуються основні теги та атрибути мови HTML.

В першому розділі розглядається, як створити web-сторінку, відформатувати шрифт: змінити шрифт, розмір, колір, зробити текст жирним, курсивом, підкресленим та інше, як застосувати до нього вирівнювання, описуються способи створення різних типів списків.

В другому розділі описується, як додати на web-сторінку графічні зображення: фотографію, малюнок, а також, як можна їх змінити, використовуючи запропонований набір тегів мови HTML.

В третьому розділі пояснюється як створити таблицю, виконати об'єднання комірок, змінювати їх розмір та заливку.

В четвертому розділі описуються зовнішні та внутрішні гіперпосилання, показано як в ролі гіперпосилань використовувати текст або графічні об'єкти.

Часто при створенні сайту виникає необхідність відкривати в одному вікні браузера одночасно декілька HTML документів. В останньому розділі розглядаються фрейми, які створені для того, щоб визначити робочі області для кожного документу, з їх допомогою можна займатися верстанням сторінки.

РОЗДІЛ 1. СТВОРЕННЯ WEB-СТОРІНКИ. ФОРМАТУВАННЯ ТЕКСТУ, СТВОРЕННЯ СПИСКІВ

HTML (Hyper Text Markup Language - мова розмітки гіпертекстових документів) - стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

Тег – це елемент стандартної мови розмітки тексту в Інтернеті — мови HTML.

Даними елементами супроводжується початок і закінчення кожної виділеної частини документа.

Теги — це одна з кількох властивостей, які користувач може додавати до файлів, щоб їх було легше знаходити та впорядковувати. Крім тегів, файли мають також інші властивості, такі як дата змінення, ім'я автора та оцінювання. Теги не входять до вмісту файлу. Натомість, теги надають інформацію щодо файлів, на підставі якої користувач може ефективніше шукати та впорядковувати файли. На відміну від інших властивостей, які зазвичай є заздалегідь визначеними, теги можна задавати довільно. Створені та додані до файлу теги стають однією з його властивостей.

Атрибути - це компоненти тега, що містять вказівки про те, як браузер повинен сприйняти і обробити тег. Атрибут записується після імені тега перед закриваючою дужкою > і складається, як правило, із пари

Для початку розглянемо, як створити web-сторінку, змінити шрифт і його розмір, зробити текст жирним, курсивом, підкресленим і про те, як його вирівнювати по краях або центру документу.

Для початку роботи необхідно відкрити блокнот або додаток "Sublimetext" і написати в ньому наступний текст:

```
<html>
<head>
<title>Моя перша сторінка</title>
</head>
<body>
Привіт свім!!!
<br>
Мене звати (тут впишіть Ваше ім'я), це моя перша
сторінка!
</body>
</html>
```

Далі, необхідно зберегти цей текст як html документ (з розширенням .html), з довільною назвою.

Приклад збереження документа: заходимо в меню "Файл", вибираємо "Зберегти як". у рядку "Ім'я файлу" пишемо: index.html, просто за замовчуванням блокнот пропонує зберегти файл з розширенням *.txt, а потрібно обрати розширення *.html

Далі потрібно відкрити цей файл за допомогою браузера (Internet Explorer, Chrom, Opera)

Параграф – це такий шматочок тексту, який в книгах зазвичай друкується з нового рядка, тим самим, виділяючи цей текст з основної маси. Книгу, розбиту на параграфи, легко читати, тому що, як правило, одному параграфу відповідає одна думка або логічна частина тексту.

Для того, щоб на сторінці сайту розбити текст на параграфи, необхідно скористатися тегом <p> - власне параграф.

Параграф має атрибут align "вирівнювання", який у свою чергу може дорівнювати тому чи іншому значенню.

Приклад:

За допомогою параграфа можна розташувати текст по центру:

```
<p align="center">Привіт свім!!!</p>
```

По лівому краю:

```
<p align="left">Привіт світ!!!</p>
```

По правому краю:

```
<p align="right">Привіт світ!!!</p>
```

Або ж обом краям документу :

```
<p align="justify">Привіт світ!!!- тут потрібний текст  
подовше, щоб ефект був добре видний при відкритті  
документу</p>
```

Внесемо зміни в першу сторінку:

```
<html>
```

```
<head>
```

```
<title>Мій перший сайт </title>
```

```
</head>
```

```
<body>
```

```
<p align="center">Привіт світ!!!</p>
```

```
<br>
```

```
<p align="justify">
```

Атрибути - це компоненти тега, що містять вказівки про те, як браузер повинен сприйняти і обробити тег. </p>

```
</body>
```

```
</html>
```

Зверніть увагу:

1) Тег <p> не може містити в собі інших параграфів, тобто писати як написано нижче - не можна, це нелогічно, тому що один параграф не може містити в собі інший.

```
<p>
```

```
<p>
```

```
</p>
```

```
</p>
```

2) Так само, не можна писати порожні теги без тексту або інших тегів.

`<p>` тут щось обов'язково має бути!!!`</p>`

3) За замовчуванням текст вирівнюється браузером по лівому краю, так що якщо так і потрібно, атрибут `align="left"` для параграфа можна не вказувати.

4) Тег `<p>` має на увазі перенесення рядка. Якщо це не потрібно, використовуйте замість тега `<p>` тег `<div>`. Цей тег є альтернативою тегу `<p>` пишеться так:

`<div align="center">Привіт свім!!!</div>`

`<div align="left">Привіт свім!!!</div>`

`<div align="right">Привіт свім!!!</div>`

`<div align="justify">Привіт свім!!!</div>`

Все те ж саме, тільки цей тег не переносить текст на наступний рядок і в нього можна вставляти тег `<p>` за принципом:

`<div>`

`<p align="left">Пишемо ліворуч</p>`

`<p align="right">Пишемо справа</p>`

`</div>`

Взагалі тег `<div>` багатфункціональний і за своєю суттю він є порожнім блоком-контейнером, який може містити в собі як текст, так і інші теги. Загалом, робота з текстом це далеко не основне завдання тега `<div>`, але про інші можливості цього тега буде описано нижче.

5) Ще одним способом вирівнювання тексту по центру є використання тега `<center>`. Будь-який зміст узятий в цей тег вирівнюється по центру екрану. Пишеться так:

`<center>Привіт свім!!!</center>`

Заголовки

У наборі тегів html мови є шість типів заголовків :

`<h1> Привіт світ!!!</h1>`

`<h2> Привіт світ!!!</h2>`

`<h3> Привіт світ!!!</h3>`

`<h4> Привіт світ!!!</h4>`

`<h5> Привіт світ!!!</h5>`

`<h6> Привіт світ!!!</h6>`

Тобто `<hx>` - тег заголовка, де значення `x` є величиною букв(може бути від одного до шести). Пам'ятайте, що після використання того або іншого заголовка відбувається перенесення рядка - на те він і заголовок.

Заголовки справа – хороша і досить зручна річ, але ними можна виділяти тільки маленькі шматочки тексту.

Для того, щоб виділити весь текст, існує тег `<font>`, у перекладі - шриффт.

Тег `<font>` окрім інших атрибутів, про які ще піде мова, має атрибут `size` - розмір.

Пишеться і виглядає це так:

`<font size="+4">Привіт світ!!!</font>`

`<font size="+2">Привіт світ!!!</font>`

`<font size="+0">Привіт світ!!!</font>`

`<font size="-1">Привіт світ!!!</font>`

`<font size="-2">Привіт світ!!!</font>`

Додамо ці теги на сторінку:

`<html>`

`<head>`

`<title>Мій перший сайт</title>`

`</head>`

`<body>`

`<center><h2>Привіт світ!!!</h2></center>`

`<br>`

`<p align="justify">`

`<font size="+1">`

Атрибути - це компоненти тега, що містять вказівки про те, як браузер повинен сприйняти і обробити

```
<font size="+2"> тег. </font> </font>
</p>
</body>
</html>
```

Спробуйте проєкспериментувати з розмірами шрифту і заголовком, підставте свої значення, змініть розташування тегів.

Використання кольору

Для надання сторіночці красивого виду не обійтися без палітри з фарбами.

Один і той же колір можна задати двома способами: використовуючи шістнадцятирічне значення кольору RGB - наприклад #008000 - зелений колір, або використовуючи константи кольору – green.

У тега є один атрибут - color.

Приклад:

```
<font color="#ff0000">Привіт свім!!!</font>
```

або

```
<font color="red">Привіт свім!!!</font>
```

У html мові використовується своя палітра фарб. Ось основні кольори, виглядають вони так:

#000000 black	#ffffff white	#ff0000 red
#ffa500 orange	#ffff00 yellow	#008000 green
#00ffff cyan	#0000ff blue	#800080 purple

Краще писати шістнадцятирічним числом, по-перше, із зрозумілих причин: не для усіх відтінків кольорів є своя назва, а по-друге, не усі браузери знають назви тих або інших фарб.

Є ще один спосіб змінити колір тексту.

Тег `<body> </body>` "тіло" - має атрибут `text` - "текст". Якщо присвоїти цьому атрибуту будь-який колір з доступної палітри, то весь текст на сторінці зафарбується, окрім тих місць, де вказаний інший колір.

У рядку де стоїть тег `<body>` пишемо так:

```
<body text="#ff8c40">
```

Тепер увесь текст у нас став помаранчевим окрім заголовка "Привіт світ!!!" який ми окремо перефарбували в червоний.

Атрибут тега `<body>` - `bgcolor` і його значення задає колір фону сторінки:

`<body bgcolor=" #40caff">` - залили усе блакитним кольором.

Приклад:

```
<html>
```

```
<head>
```

```
<title>Мій перший сайт </title>
```

```
</head>
```

```
<bodytext="#484800" bgcolor="#e8e8e8">
```

```
<center>
```

```
<h2>
```

```
<fontcolor="#008000">Привіт світ!!!</font>
```

```
</h2>
```

```
</center>
```

```
<p align="justify">
```

`<font size="+1">` Атрибути - це компоненти тега, що містять вказівки про те, як браузер повинен сприйняти і обробити `<font size="+2"color="#ff0000">` тег. `</font>`

```
</font>
```

</p> </body> </html>

Зверніть увагу на те, як пишеться код, якщо в тегу є присутні два і більше атрибути. У рядку <body text="#ff207b" bgcolor=" #1a77f0"> у тега <body> два атрибути text і bgcolor, ми просто пишемо їх підряд через пропуск, не розділяючи ніякими іншими знаками.

Стиль тексту

Нові теги:

Напівжирний текст

<i></i>

Похилий текст

<u></u>

Підкреслений текст

<strike></strike>

Перекреслений

<s></s>

Перекреслений(другий

варіант, від

першого нічим не відрізняється)

<tt></tt>

Моноширинний шрифт

<small></small>

Малий

<big></big>

Великий

Верхній^{індекс}

Нижній_{індекс}

Текст ув'язнений між цими відкриваючими і закриваючими тегами, набуває потрібного нам стилю.

Приклад:

<html>

<head>

<title>Стилі тексту</title>

</head>

<body>

<big><u>Наукова стаття.</u></big>

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

Основи інформаційних технологій. HTML для початківців

Якщо розбавити дистильовану воду H_2O сорока відсотками етилового спирту C_2H_5OH то вийде *рідина* в 40%

Уперше цю пропорцію<s>запропонував</s><tt>Дмитро Іванович Менделєєв.</tt>

<small>Вживання такої рідини шкодить вашому здоров'ю.</small>

</body>

</html>

Шрифт

Для того, щоб змінити шрифт документу, необхідно дати вказівку браузеру. Для цього використовуємо тег і його атрибут face.

Пишеться так:

Цей рядок буде написаний за допомогою шрифту Arial

Приклад:

<html>

<head>

<title>Використання різних шрифтів</title>

</head>

<body>

Мало хто знає, як багато потрібно знати для того, що б знати, як мало ми знаємо.


```
<font face="times new roman">Мало хто знає, як багато  
потрібно знати для того, що б знати, як мало ми  
знаємо.</font>
```

```
<br>
```

```
<font face="comic sans ms">Мало хто знає, як багато  
потрібно знати для того, що б знати, як мало ми  
знаємо.</font>
```

```
<br>
```

```
</body>
```

```
</html>
```

Браузер використовує бібліотеку шрифтів, встановлену на комп'ютері користувача, і якщо раптом вказаного Вами шрифту в цій бібліотеці не опиниться, то він замінить його на той, який є присутнім.

Існує тег `<pre>`, текст ув'язнений в цей тег виводиться браузерами на екран в тому вигляді в якому він був набраний, тобто з усіма пропусками і перенесеннями рядків

Приклад:

```
<html>
```

```
<head>
```

```
<title>Пропуски і перенесення рядка</title>
```

```
</head>
```

```
<body>
```

```
<pre>
```

Інтернет – корисна справа,

Розважальна і цікава,

Пізнавальна, бо про все

Інформацію несе.

```
</pre>
```

```
</body>
```

```
</html>
```

Створення списків

Маркірований список задається тегом `<ul>`. Будь-який список складається з елементів, "пунктів". Кожен елемент у свою чергу задається тегом `<li>`, після якого йде текст потрібного нам "пункту". Для тега `<li>` закриваючий тег `</li>` необов'язковий.

Разом ці теги набувають наступного вигляду:

```
<ul>
<li> пункт перший.
<li> пункт другий.
<li> пункт третій.
</ul>
```

Приклад маркірованого списку :

```
<html>
<head>
<title> Список</title>
</head>
<body>
Купити продукти додому:
<ul>
<li> 2 кг картоплі
<li> 1 булку житнього чорного хліба
<li> пачку макаронів
<li> 1 кг шліфованого рис)
<li> 1 дес. курячих яєць.
</ul>
</body>
</html>
```

Теги `<li>` і `<ul>` мають атрибут `type`, який присуджує елементу списку або ж усьому списку певний стиль.

Може мати три значення:

disk - гурток, диск(за замовчуванням);

circle - порожнистий круг;

square - квадрат.

Приклад:

```
<html>
<head>
<title>стили списку</title>
</head>
<body>
```

У цьому списку кожен елемент має свій стиль:

```
<ul>
<li type="disk"> диск(за замовчуванням )
<li type="circle">порожнє коло
<li type="square">квадрат
</ul>
```

А тут стиль заданий усьому списку

```
<ul type="circle">
<li> Пункт 1
<li> Пункт 2
<li> Пункт 3
</ul>
</body>
</html>
```

Впорядковані списки

Впорядкований або нумерований список задається тегом `<ol>`, так само як і в маркірованому списку елемент списку прописується тегом `<li>`.

Побудова коду повністю схожа з попереднім списком.

Приклад:

```
<html>
<head>
<title>Нумерований список</title>
</head>
<body>
```

Купити товари в наступному порядку:

```
<ol>
<li>хліб
<li>сметана
<li>морозиво
</ol>
</body>
</html>
```

А ось атрибут *type* у поєднанні з впорядкованим списком може мати наступні значення:

A - заголовні букви;
a - рядкові букви;
I - заголовні римські цифри;
i - рядкові римські цифри;
1 - арабські цифри(за замовчуванням).

Приклад:

```
<html>
<head>
<title>Стилі впорядкованого список</title>
</head>
<body>
Арабські цифри
<oltype="1">
<li>По-перше
<li>По-друге
</ol>
Рядкові букви
<oltype="a">
<li>По-перше
<li>По-друге
</ol>
Заголовні букви
```

```
<ol type="A">  
<li>По-перше  
<li>По-друге  
<li>По-третє  
</ol>
```

Рядкові римські цифри

```
<ol type="i">  
<li>По-перше  
<li>По-друге  
<li>По-третє  
<li>По-четверте  
</ol>
```

Заголовні римські цифри

```
<ol type="I">  
<li>По-перше  
<li>По-друге  
<li>По-третє  
<li>По-четверте  
</ol>
```

```
</body>
```

```
</html>
```

У впорядкованому списку є ще один атрибут *start*. Його числове значення говорить про те, з якого номера слід будувати впорядкований список.

Приклад:

```
<html>  
<head>  
<title>Початок впорядкованого списку</title>  
</head>  
<body>  
<ol type="1" start="24">  
<li>Відразу переходимо до двадцять четвертому  
пункту!!  
<li>Йдемо далі
```

```
<li>I далі
```

```
</ol>
```

Аналогічно можна "стартувати" при будь-якому стилі впорядкованого списку

```
<ol type="I"start="8">
```

```
<li>Відразу переходимо до восьмого пункту.
```

```
<li>Йдемо далі
```

```
<li>I далі
```

```
</ol>
```

```
</body>
```

```
</html>
```

Списки визначень

Зі списком визначень справа йде. Задається цей вид списку тегом `<dl>`. Пункти списку визначень розмічаються тегом `<dt>`, а визначення цих пунктів тегом `<dd>`. Все разом пишеться за наступною схемою:

```
<dl>
```

```
<dt>
```

```
<dd>
```

```
</dl>
```

Приклад:

```
<html>
```

```
<head>
```

```
<title>Список визначень</title>
```

```
</head>
```

```
<body>
```

```
<dl>
```

```
<dt>Слово коса може мати наступні визначення:
```

```
<dd>сільськогосподарський інструмент
```

```
<dd>хитра дівоча зачіска
```

```
<dd>мілина річки
```

```
<dt>Слово ключ теж має декілька значень:
```

```
<dd>гайковий
```

```
<dd>джерело, джерело  
</dl>  
</body>  
</html>
```

Маркіровані переліки `<ul>` використовуються тоді, коли порядок дотримання пунктів не важливий, і робляться вони для того, щоб викладена інформація сприймалася легше. А нумеровані списки `<ol>` навпаки, застосовуються тоді, коли важливий саме порядок дотримання пунктів, наприклад - коли список є викладом послідовності дій.

РОЗДІЛ 2. ВСТАВКА ГРАФІЧНИХ ОБ'ЄКТІВ.

В цьому розділі описується, як додати на web-сторінку графічні зображення: фотографію, малюнок, а також, як можна їх змінити, використовуючи запропонований набір тегів мови HTML.

Шлях до файлу

Припустимо, Ви пишете сайт про себе і хочете вставити в нього свою фотографію, щоб люди зайшли на сторіночку і змогли не лише читати Ваші рукописи, але й помилуватися Вашим портретом.

Копіюємо фотографію, яка знаходиться на Вашому жорсткому диску і вставляємо її в ту теку(директорію), де вже лежить заготовка майбутньої сторінки, ну тобто туди ж, куди зберігаєте html документ. Для того, щоб вставити її в сторінку, до неї треба вказати шлях: ``, де `foto.jpg` - це назва Вашої фотографії в даному випадку. Оскільки фото лежить поряд з html документом, шлях до неї ми не вказуємо.

Ряд прикладів де шлях вказується:

`` - такий запис має на увазі, що в директорії де розташований наш html документ є тека `myfoto` в якій розташований файл `foto.jpg`

`` - означає поряд з документом розташована тека `myfoto`, в ній ще одна тека з назвою `graphics`, і вже в ній потрібна нам фотографія `foto.jpg` яку треба викласти для загального огляду.

`` - означає, що фото розміщене на рівень вище від документу

`` - відповідно на два рівні вище, скільки поставите./ настільки і підніміться.

Так само можна вказуючи місце фотографії, посилатися на той або інший інтернет ресурс:

Приклад:

<html>

<head>

<title>Моя перша сторіночка з фото</title>

</head>

<body text="#484800" bgcolor="#e8e8e8">

<center>

<h2> Привіт свім!!!</h2>

</center>

<p align="justify">

Атрибути - це компоненти тега, що містять вказівки про те, як браузер повинен сприйняти і обробити меє.

</p>

</body>

</html>

Як видно з цього прикладу, фотографія з'явилася, але разом зі своєю появою зрушила текст.

Атрибути тега :

 - фото зліва від тексту

 - фото праворуч від тексту

 - текст вищий за фото

 - текст нижчий фото

 - ну і відповідно текст посередині

Окрім align тег має ще ряд атрибутів.

Піксель - це елементарна неділима одиниця зображення. Кожен піксель має свої координати, наприклад, самий верхній

лівий піксель на моніторі має координати $x=1$, $y=1$, а самий нижній правий залежно від дозволу монітора. $x=800$, $y=600$ - буде відповідно при дозволі 800 на 600 точок. Втім, ця інформація згодиться нам потім, а зараз треба засвоїти, що усі відстані в графічних зображеннях міряються пікселями. так картинка довжини 800 пікселів і шириною в 600 пікселів при вказаному дозволі заповнить увесь екран.

Додаткові атрибути:

`` - атрибут `vspace` задає відстань по вертикалі від малюнка до тексту, в даному випадку ми задали відстань в 15 пікселів

`` - відстань по горизонталі відповідно

`` - ширина безпосередньо самого зображення

`` - висота зображення. Якщо атрибути `width` і `height` не використати, то ширина і висота зображення за замовчуванням дорівнюватиме реальним її розмірам, без яких або спотворень.

`` - бордюр, рамка навколо зображення і її товщина в пікселях.

`` - `bordercolor` - це колір рамки.

`` - атрибут `alt` - цей опис зображення. Якщо навести курсор на наше фото і потримати його там декілька секунд, вискочить напис - Це моє фото!!!

`` - альтернатива `alt` в даному випадку.

Також зображення можна зробити фоном сторінки. Для цього використовуємо атрибут `background` "фон":

`<body background="foto.jpg">`

Вирівнювання малюнка

Розміщення фотографії в потрібному місці сторінки відбувається за принципом вирівнювання тексту.

Приклад:

```
<html>
<head>
<title>Розташування зображення</title>
</head>
<body>

<center></center>
<div align="left"></div>
<br>
<div align="right"></div>
<br>
<div align="center"></div>
</body>
</html>
```

РОЗДІЛ 3. СТВОРЕННЯ ТАБЛИЦЬ

Окрім інших об'єктів у свій сайт Ви можете вставити таблиці. Вони мають велику значимість при створенні сайту. За допомогою таблиці можна не лише викласти ту або іншу інформацію, тарифну сітку або графік чергувань приміром, але і, узявши її за основу, повністю побудувати на ній свій сайт. Таблиці іноді незамінні при верстанні сторінки. А зараз давайте навчимося її малювати.

Тег `<table>` задає початок і кінець таблиці, будь-яка таблиця, як відомо, складається з рядків і стовпців, для цього є ще два теги:

`<tr>` - рядок таблиці

`<td>` - стовпець таблиці

Разом ці теги пишуться так:

```
<table>
```

```
<tr>
```

```
<td>оцепедок</td>
```

```
</tr>
```

```
</table>
```

Такий запис це найменша таблиця, в ній всього один рядок, що містить один стовпець.

Поставимо перед собою завдання намалювати таблицю з трьох рядків і трьох стовпців, а разом згадаємо атрибут `border` "рамка", який додасть наочності.

Приклад:

```
<html>
```

```
<head>
```

```
<title>Таблиця</title>
```

```
</head>
```

```
<body>
```

```
<table border="1">
```

```
<tr>
<td>рядок1 комірка1</td>
<td>рядок1 комірка2</td>
<td>рядок1 комірка3</td>
</tr>
<tr>
<td>рядок2 комірка1</td>
<td>рядок2 комірка2</td>
<td>рядок2 комірка3</td>
</tr>
<tr>
<td>рядок3 комірка1</td>
<td>рядок3 комірка2</td>
<td>рядок3 комірка3</td>
</tr>
</table>
</body>
</html>
```

Об'єднання комірок

Часто при роботі з таблицями виникає необхідність об'єднати ті або інші комірки в одну.

Цю задачу вирішують атрибути *colspan* і *rowspan*:

colspan - визначає яку кількість стовпців необхідно об'єднати;

rowspan—визначає яку кількість рядів необхідно об'єднати.

Припустимо що в нашому прикладі нам необхідно об'єднати в третьому рядку другу і третю комірки, для цього атрибуту *colspan* привласнюємо значення 2 (розтягнути на два стовпці) і вставляємо його в потрібне місце.

Приклад:

```
<html>
<head>
```

```
<title>Таблиця</title>
</head>
<body>
<table border="1">
<tr>
<td>строка1 комірка1</td>
<td>строка1 комірка2</td>
<td>строка1 комірка3</td>
</tr>
<tr>
<td>строка2 комірка1</td>
<td>строка2 комірка2</td>
<td>строка2 комірка3</td>
</tr>
<tr>
<td>строка3 комірка1</td>
<td colspan="2">строка3 комірка2</td>
</tr>
</table>
</body>
</html>
```

Тепер необхідно об'єднати весь перший стовпець, використовуючи атрибут `rowspan`:

```
<html>
<head>
<title>Таблиця</title>
</head>
<body>
<table border="1">
<tr>
<td rowspan="3">рядок1 комірка1</td>
<td>рядок1 комірка2</td>
<td>рядок1 комірка3</td>
```

```
</tr>
<tr>
<td>рядок2 комірка2</td>
<td>рядок2 комірка3</td>
</tr>
<tr>
<td colspan="2">рядок3 комірка2</td>
</tr>
</table>
</body>
</html>
```

Розміри таблиці і комірок за замовчуванням обмежені вставленим в неї текстом. Згадуємо про атрибути *width* - ширина і *height* - висота, які ми використали для розтягання малюнків, вони так само можуть бути застосовані до тегів `<table>`, `<tr>` і `<td>`. У прикладі нижче задані розміри таблиці і окремих її комірок, і вся таблиця вирівняна по центру тегом `<center>`.

Приклад:

```
<html>
<head>
<title>Таблиця</title>
</head>
<body>
<center>
<table border="1"width="640" height="480">
<tr>
<td rowspan="3"width="150">строка1 ячейка1</td>
<td height="30">срядок1 комірка2</td>
<td>рядок1 комірка3</td>
</tr>
<tr>
<td height="30">рядок2 комірка2</td>
<td>рядок2 комірка3</td>
```

```
</tr>
<tr>
<td colspan="2">рядок3 комірка2</td>
</tr>
</table>
</center>
</body>
</html>
```

Зверніть увагу на те, що довжина і ширина задані не для усіх комірок. Таблиця буде вирівняна по найширших і довших осередках.

Значення атрибутів *width* і *height* в таблиці можуть вказуватися не лише в пікселях, але і у відсотках

Приклад:

```
<html>
<head>
<title>Таблиця</title>
</head>
<body>
<center>
<table border="1" width="640" height="480">
<tr height="25%">
<td width="15%">рядок1 комірка1</td>
<td width="25%">рядок1 комірка2</td>
<td width="60%">рядок1 комірка3</td>
</tr>
<tr height="50%">
<td>рядок2 комірка1</td>
<td>рядок2 комірка2</td>
<td>рядок2 комірка3</td>
</tr>
<tr height="25%">
```

```
<td>рядок3 комірка1</td>
<td>рядок3 комірка2</td>
<td>рядок3 комірка3</td>
</tr>
</table>
</center>
</body>
</html>
```

В даному прикладі від загальної ширини таблиці в 640 пікселів відводиться 15% першому блоку, 25% другому і 60% третьому. А по висоті з 480 пікселів по 25% першому і третьому рядку, а добру половину місця другої. В сумі потрібно щоб виходило 100%.

Верстання сторінки з використанням таблиці

Таблиця – це хороший і головне зручний інструмент для верстання сторінки.

Будь-який елемент таблиці може служити самостійною місткістю для наповнення іншими тегами і текстовим змістом, а також мати ті або інші індивідуальні властивості - атрибути.

```
<html>
<head>
<title>Верстання сторінки</title>
</head>
<body>
<center>
<table border="1" width="640" height="480">
<tr>
<td colspan="5" height="30"><center>Заголовок</center>
</td>
</tr>
<tr>
<td height="30" width="20%"> <center>кнопка1</center>
</td>
```

```
<td width="20%"><center>кнопка2</center></td>
<td width="20%"><center>кнопка3</center></td>
<td width="20%"><center>кнопка4</center></td>
<td width="20%"><center>кнопка5</center></td>
</tr>
<tr>
<td valign="top">зміст:</td>
<td colspan="4"><center>кнопка</center></td>
</tr>
</table>
</center>
</body>
</html>
```

Приклад з додаванням кольору:

```
<html>
<head>
<title>Таблиця і фарби</title>
</head>
<body>
<center>
<table width="300" height="300">
<tr>
<td bgcolor="#c0e4ff">1</td>
<td bgcolor="#c5ffa0">2</td>
<td bgcolor="#c0e4ff">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5">5</td>
<td bgcolor="#c5ffa0">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff">7</td>
```

```
<td bgcolor="#c5ffa0">8</td>
<td bgcolor="#c0e4ff">9</td>
</tr>
</table>
</center>
</body>
</html>
```

Вирівнювання тексту

Існує атрибут *valign* - вертикальне вирівнювання. До цього моменту нам був знайомий атрибут *align* - горизонтальне вирівнювання.

Приклад:

```
<html>
<head>
<title>Вирівнювання в таблиці</title>
</head>
<body>
<center>
<table width="300" height="300">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
<td bgcolor="#c5ffa0" valign="top">
<center>2</center></td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"><center>5</center></td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff" valign="bottom">7</td>
```

```
<td bgcolor="#c5ffa0" valign="bottom"> <center> 8 </center>
</td>
<td bgcolor="#c0e4ff" align="right" valign="bottom">9</td>
</tr>
</table>
</center>
</body>
</html>
```

Атрибут *Cellspacing* - задає відстань в пікселях між елементами таблиці. Задавши значення `cellspacing="0"` можна позбавитися від "проміжку" між комірками.

Приклад, збільшимо "проміжок":

```
<html>
<head>
<title>Відстань між комірками</title>
</head>
<body>
<center>
<table width="300" height="300"cellspacing="15">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
<td bgcolor="#c5ffa0" valign="top">
<center>2</center></td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"><center>5</center></td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff" valign="bottom">7</td>
```

```
<td bgcolor="#c5ffa0" valign="bottom"> <center>
8</center></td>
<td bgcolor="#c0e4ff" align="right" valign="bottom">
9</td>
</tr>
</table>
</center>
</body>
</html>
```

Атрибут *cellpadding* (не плутайте з *cellspacing*) - в пікселях задає поля осередків(відступ від меж осередків до тексту).

Приклад:

```
<html>
<head>
<title>Поля таблиці</title>
</head>
<body>
<center>
<table width="300" height="300" cellspacing="0"
cellpadding="25">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
<td bgcolor="#c5ffa0" valign="top">
<center>2</center></td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"><center>5</center></td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
</tr>
```

```
<td bgcolor="#c0e4ff" valign="bottom">7</td>
<td bgcolor="#c5ffa0" valign="bottom"> <center>
8</center> </td>
<td bgcolor="#c0e4ff" align="right" valign="bottom">
9</td>
</tr>
</table>
</center>
</body>
</html>
```

Вкладені таблиці

Приклад:

```
<html>
<head>
<title>Вкладена таблиця</title>
</head>
<body>
<center>
<table width="600" height="400" border="1">
<tr>
<td colspan="2"><center>Заголовок</center></td>
</tr>
<tr>
<td valign="top">зміст:</td>
<td valign="top">
<center>
<br><br>
У одну комірку вкладена ціла таблиця!!
<br><br><br>
<table width="200" height="200" cellpadding="15">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
```

```
<td bgcolor="#c5ffa0" valign="top">
<center>2</center> </td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"> <center>5</center> </td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff" valign="bottom">7</td>
<td bgcolor="#c5ffa0" valign="bottom">
<center>8</center> </td>
<td bgcolor="#c0e4ff" align="right"
valign="bottom">9</td>
</tr>
</table>
</center>
</td>
</tr>
</table>
</center>
</body>
</html>
```

РОЗДІЛ 4. ПОСИЛАННЯ

У цьому розділі розглянемо, які бувають посилання і як їх правильно використовувати.

Щоб браузеру нормально працював, йому необхідно знати: точну назва документу, шлях до документу, і місце, де його відкрити.

На даний момент за допомогою блокнота ми створили HTML документ. В цьому документі потрібно створити посилання на інший документ, якого у нас ще немає. Перш ніж на нього посилатися, його треба створити:

Відкриваємо блокнот.

Пишемо код html мовою (наприклад сторіночку з рядом фотографією);

Зберігаємо його як html сторіночку в ту ж робочу теку, де вже є створений нами перший документ, але під іншим ім'ям.

Текстові посилання

Розглянемо тег `<a>` (від *anchor* - якір) у нього можна вкласти текст або малюнок, які стануть посиланням на ті або інші документи. Атрибут тега `<a>` `href` задає ім'я і шлях до документу на який вказує посилання.

Все разом пишеться так:

`<a href="primer.html">Тут мої фото!</a>`,

де `primer.html` це ім'я другого html документу, а напис "Тут мої фото!" це шматочок тексту з першого файлу.

`<a href="stranica/primer.html">Тут мої фото!</a>` - такий запис має на увазі, що в директорії, де розташований наш перший html документ є тека `stranica` в якій розташований файл `primer.html`

`<a href="/primer.html">Тут мої фото!</a>` - це означає, що файл `primer.html` розміщений на рівень вище від документу

`<a href="http://www.site.ru/primer.html">` Тут мої фото!`</a>` - документ розташований на сайті www.site.ru.

Нижче наведений приклад відразу двох документів в яких прописані посилання, що вказують один на одного.

Приклад:

Файл `index.html`:

```
<html>
<head>
<title>Робимо посиланням шматочок тексту</title>
</head>
<body>
<div align="center">
<br><br><br><b>Оберіть один з варіантів:</b>
<br><br><br>
<a href="primer1.html">на право</a>чи
<a href="primer2.html">на ліво</a>?
</div>
</body>
</html>
```

Файл `primer1.html`:

```
<html>
<head>
<title>Перешли по посиланню сюди</title>
</head>
<body>
<br><br><br><br>
<div align="center"><font size="+2">На право ліс</font>
</div>
<br><br><br>
<div align="center">
<a href="index.html">повернутися</a> </div>
</body>
```

</html>

Файл *primer2.html*:

<html>

<head>

<title>Перешли по посиланню сюди</title>

</head>

<body>

<div align="center">На ліво річка

</div>

<div align="center">

повернутися </div>

</body>

</html>

З прикладу видно, що посилання виділяються кольором, за замовчуванням синім, а червоним - вже відвідане посилання. Ці кольори можна змінити з допомогою вже добре відомого відкриваючого тега < body > і його атрибутів.

link - колір посилання.

alink - колір натиснутого, активного посилання.

vlink - колір відвіданого посилання.

Пишеться так:

<body link="#008000" alink="#ff0000" vlink="#ffff00">

При необхідності можна примусово виділяти кольором як усе посилання, так і окремі його частини (фрази слова букви) тегом і його атрибутом *color*. Втім, це стосується не лише кольору. Так само окремо можна задати розмір, стиль і шрифт тексту. Але пам'ятаєте, що маніпуляції з кольором треба

проводити усередині тега `<a>ось тут</a>` а не поза ним, інакше колір посилання буде за замовчуванням .

Приклад:

Файл index.html:

```
<html>
<head>
<title>Веселка</title>
</head>
<body link="#008000" alink="#ff0000" vlink="#ffff00">
<center>
<h3>Подивіться як розміщуються кольори у
веселці</h3>
<br>
<a href="primer 2.html">
<font size="+1" color=#ff0000>червоний</font>
<font size="+2" color=#ff8c40>помаранчевий</font>
<font size="+3" color=#ffff00>жовтий</font>
<font size="+3" color=#008000>зелений</font>
<font size="+2" color=#40c0ff>блакитний</font>
<font size="+1" color=#0000ff>синій</font>
<font size="+1" color=#800080>фіолетовий</font>
</a>
</center>
</body>
</html>
```

Файл primer.html:

```
<html>
<head>
<title>Веселка</title>
</head>
<body link="#008000" alink="#ff0000" vlink="#ffff00">
```

```
<center>
<font size="+3">
ВЕСЕЛКА
</font>
<br><br><br>
<a href="index.html">повернутися на головну</a>
</center>
</body>
</html>
```

Малюнок-посилання

Посиланням може бути не лише текст, але і малюнок. Тут принцип такий же як і в текстовому посиланні, просто замість тексту ми укладаємо малюнок, який хочемо зробити посиланням.

Пишеться так:

```
<a href="primer 3.html"></a>
```

Перш ніж навести черговий приклад, розглянемо, як при переході по посиланню відкрити документ в новому вікні браузера. До цього він у нас відкривався в поточному вікні, що не завжди зручно. Вирішує цю проблему атрибут *target* (мета) і його значення *_blank*.

Пишеться так:

```
<a href="primer 3.html" target="_blank">відкрити в новому вікні</a>
```

Приклад:

Файл index.html:

```
<html>
<head>
<title>кнопка</title>
</head>
```

```
<body>
<center>
<h1> Ні в якому випадку не натискайте на цю
кнопку!!!</h1>
<br>
<a href="primer 3.html" title="Не натискати!!!"
target="_blank">

</a>
</center>
</body>
</html>
```

Файл primer.html:

```
<html>
<head>
<title>нідсумок</title>
</head>
<body>
<div align="center"><font size="+3">
Ха-ха, це був жарм.</font></div>
</body>
</html>
```

По-перше, тут використовуються вже знайомі атрибути опису `title="Не натискати!!!"` для тега `<a>` і одночасно `alt="Не натискати!!!"` для тега `<img>`.

По-друге, у прикладі використаний атрибут тега `<img border="0">` - рамка малюнка.

0 - це означає вимкнути рамку, оскільки вона псує увесь вид сторінки в цьому прикладі, але бувають випадки коли рамки малюнка цілком актуальні і так би мовити вписуються в сторінку. Тут треба пам'ятати, що коли малюнок є посиланням,

його рамка забарвлюється в кольори посилання : *link* - колір посилання, *alink* - колір натиснутого, активного посилання, *vlink* - колір відвіданого посилання тега `<body>`.

Посилання на e-mail

Для того що б зробити текст або малюнок посиланням на поштову скриньку, його треба вкласти в тег `<a>`, але не простий, а з використанням *mailto*.

Пишеться це так:

`<a href="mailto:karlson@kruha.ua"> Напишіть мені лист. </a>`

Цей незвичний запис говорить що, кликнувши за текстом посилання "Напишіть мені лист.", відвідувач сайту потрапить у свою поштову програму, яка видасть бланк для відправки листа, де в рядку "Кому:" вже буде вказана потрібна нам поштова скринька *meni@gmail.com*

Приклад:

```
<html>
<head>
<title>e - mail</title>
</head>
<body>
<center><h2>
<a href="mailto: meni@gmail.com "> Напишіть мені
лист. </a>
</h2> </center>
</body>
</html>
```

Окрім цього в поштовому бланку можна заздалегідь прописати наступні речі:

?subject= - тема листа

&Body= - текст повідомлення

&cc= meni1@gmail.com, meni2@gmail.com - копії листа

&bcc= meni3@gmail.com, meni4@gmail.com - приховані копії листа.

Адреси ящиків для копій і прихованих копій листа пишуться через кому.

Приклад:

```
<html>
<head>
<title>e - mail</title>
</head>
<body>
<center><h2>
<a href="mailto: meni@gmail.com?subject=Як  ваші
справи?. &
Body=Текст листа &cc= meni@gmail.com
&bcc= meni3@gmail.com ">Напишіть мені листа. </a>
</h2></center>
</body>
</html>
```

Закладки

Закладки або **якорі** - це особливий вид посилань. Ці посилання, як правило, не ведуть до документа, а призначені для навігації усередині сторінки.

Припустимо, на сторінці в зручному місці знаходиться деякий зміст або меню за принципом:

Розділ1

Розділ2

Розділ3

Далі йде великий текст з цими главами. Щоб відвідувач сторінки натиснувши на одну з цих глав і перейшов в потрібне місце тексту, треба присвоїти індивідуальне ім'я кожній главі.

Приклад:

`<a name="rozdil1">Розділ1 </a>`

`<a name="rozdil2">Розділ2 </a>`

`<a name="rozdil3">Розділ3 </a>`

А тепер власне прописуємо на них посилання в меню, змісті:

`<a href="#rozdil1">Розділ1</a>`

`<a href="#rozdil2">Розділ2</a>`

`<a href="#rozdil3">Розділ3</a>`

Хоча закладки використовують для навігації усередині сторінки, на них все ж можна посилатися з інших сторінок або сайтів

`<a href="primer.html#glava1">- йдемо наРозділ1 з іншої сторінки сайту</a>`

`<a href="http://www.site.ru/ primer.html#glava2">- розмістивши таке посилання на будь-якому іншому сайті ми згідно з вказаним шляхом спочатку потрапляємо на наш сайт www.site.com дамо там документ primer.html і відразу розпочинаємо читати главу2</a>`

РОЗДІЛ 5. ФРЕЙМИ

Часто при створенні сайту виникає необхідність відкривати в одному вікні браузера одночасно декілька HTML документів. Так от фрейми створені для того, щоб визначити робочі області для кожного документу. Крім того, фрейми - це хороший інструмент, за допомогою якого можна займатися верстанням сторінки, вони служать гідною альтернативою табличному способу верстання сторінки.

Припустимо нам необхідно в одному вікні браузеру відкрити відразу три HTML документу і розташувати їх приміром ось у такий спосіб:

Документ 1

Документ 2

Документ 3

Спершу необхідно створити три окремих html документи які власне і будуть відкриватись в одному вікні. Нехай перший документ нестиме в собі графічне зображення і виконуватиме роль логотипу, назвемо його logotype.html, другим документом буде деякий зміст. назвемо menu.html, і третім - документ з купою тексту. text.html. Звичайно Ви можете придумати свої назви, як і зміст документів.

Отже, є три файли logotype.html, menu.html і text.html, які нам необхідно розмістити під одним дахом.

Пишемо "дах", головний документ, до якого підключатимемо наші файли. Так як він буде головним, привласнюємо йому ім'я index.html:

```
<html>
<head>
<title>фрейми</title>
</head>
<body>
</body>
</html>
```

Структура фрейм документу має ось такий вигляд:

```
<html>
<head>
<title>фрейми </title>
</head>
<frameset>
</frameset>
</html>
```

Відсутність тега `<body>` компенсує новий тег `<frameset>` - встановити фрейм або набір фреймів.

Тег `<frameset>` має атрибути `rows` і `cols` - ці атрибути вказують браузеру на те, як слід розміщувати фрейми у вікні браузера:

`rows` - горизонтально



`cols` - вертикально



У даному випадку нам потрібне горизонтальне розміщення фреймів. Пишемо так:

```
<html>
<head>
<title>фрейми </title>
</head>
<framesetrows="15%,15%, 70%">
</frameset>
</html>
```

Відсотки після знаку рівності - це розміри вікон-фреймів в одному великому вікні браузеру. Тут передбачається розміщення трьох вікон - отже і значень через кому теж три. Як і у випадку з елементами таблиці, розміри фреймів можна вказувати у відсотках від загальної площі і в пікселях.

Ось декілька прикладів написання:

`<frameset rows="15%,15%, 70%">` - такий запис говорить про те, що буде три горизонтальні віконця де останнє займе 70% площі вікна браузеру, а перші два по 15%.

`<frameset cols ="200, 240,200">` - тут три вікна розташовані вертикально, ширина яких вказана в пікселях.

`<frameset rows="100*,180">` - такий запис означає, що перше і третє вікно матимуть ширину 100 і 180 пікселів відповідно, а другий фрейм займе всю ту площу, що залишилась.

Підключення документів

Тег `<frame>` і його атрибут `src` вкаже браузеру шлях до html документу, який слід відкрити у відведеному для нього місці. У нас є три окремі документи `logotype.html`, `menu.html` і `text.html`. Тепер до кожного треба прописати шлях (за умови, що всі чотири файли знаходяться в одній теці і шляхи до них мають найпростіший запис за типом: `<frame src="logotype.html">`)

Приклад:

Файл `index.html`

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset rows="15%,15%, 70%">
<frame src="logotype.html">
<frame src="menu.html">
<frame src="text.html">
</frameset>
</html>
```

Файл logotype.html

```
<html>
<head>
<title>фрейми</title>
</head>
<body>

</body>
</html>
```

Файл menu.html

```
<html>
<head>
<title>фрейми</title>
</head>
<body bgcolor="#e8e8e8" background="fon 1.jpg">
<font face="Monotype Corsiva" size="6"
color="#ff0000">Меню:</font><br>
<font face="Comic Sans MS" size="3">
<a href="text.html">Суп з шампінйонів</a><br>
<a href="text 1.html">Боби в горцику по-
італійськи</a><br>
<a href="text 2.html">Австралійський літній
салат</a><br>
... ..
</font>
</body>
</html>
```

Файл text.html

```
<html>
<head>
<title>фрейми</title>
```

```
</head>
<body bgcolor="#f5f5f5">
<center><font face="Comic Sans MS" size="3">Суп з
шампінйонів</font></center>
<br>
куна тексту.
</body>
</html>
```

У цьому прикладі підключено всі чотири HTML документи, де index.html є основним, а інші три, це сторінки, що підключаються, кожна з яких в принципі може працювати автономно і містити в собі все що завгодно: картинки, таблиці, тексти, посилання. Усі вікна розташовані горизонтально, помінявши атрибут rows на cols можна розташувати їх у вертикальному порядку. А як бути, якщо вимагається розмістити наші вікна у такому вигляді:



Вихід є. Розберемо на прикладах.

Перший випадок - два рядки, де другий поділений на два стовпці. Записується це так:

`<frameset rows="20%,80%">`- ділимо вікно браузера на два рядки

`<frame src="logotype.html">`- в перший рядок завантажуюємо логотип

`<frameset cols="30%,70%">`- а другий рядок ділимо на два стовпці

`<frame src="menu.html">`- в першому буде меню

`<frame src="text.html">`- а в другому багато тексту

`</frameset>`- закриваємо тег ділення на стовпці

`</frameset>`- закриваємо тег ділення на рядки.

Говорити стовпці і рядки – невірно, правильно говорити горизонтальні і вертикальні фрейми.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset rows="15%,85%">
<frame src="logotype.html">
<frameset cols="20%,80%">
<frame src="menu.html">
<frame src="text.html">
</frameset>
</frameset>
</html>
```

Другий випадок - два стовпці, і другий поділений на два рядки. Записується це так:

```
<frameset cols="30%,70%">- ділимо вікно на два стовпці
<frame src="menu.html">- в першому буде зміст
<frameset rows="20%,80%">- а другий ділимо на два
рядки
```

```
<frame src="logotype.html">- логотип
<frame src="text.html">- основний текст
</frameset>- закриваємо ділення на рядки
</frameset>- закриваємо ділення на стовпці.
```

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="20%,80%">
```

```
<frame src="menu.html">
<frameset rows="15%,85%">
<frame src="logotype.html">
<frame src="text.html">
</frameset>
</frameset>
</html>
```

Третій випадок трохи складніший - три стовпці, причому в другий стовпець по суті поміщений наш перший випадок:

`<frameset cols="*,800*">`- ділимо вікно на три стовпчики(зверніть увагу другий стовпець займає рівно 800 пікселів, а два крайніх точного розміру не мають і поділять місце, що залишилося,навпіл)

`<frame src="dekor.html">`-в перший стовпець завантажуюмо html документ, який виконуватиме суто декоративну функцію.

```
<frameset rows="15%,85%">
<frame src="logotype.html">
```

`<frameset cols="20%,80%">`-в другий стовпець вставляємо наш "перший випадок"

```
<frame src="menu.html">
<frame src="text.html">
</frameset>
</frameset>
```

`<frame src="dekor.html">`- в третій стовпець завантажуюмо все той же файл з декораціями

```
</frameset>
```

- закриваємо

Приклад:

```
<html>
<head>
<title>фпейму</title>
</head>
<frameset cols="*,800*">
```

```
<frame src="dekor.html">
<frameset rows="120*">
<frame src="logotype.html">
<frameset cols="200, 600">
<frame src="menu.html">
<frame src="text.html">
</frameset>
</frameset>
<frame src="dekor.html">
</frameset>
</html>
```

Приводимо фрейми в охайний вид

У останньому прикладі перше, що впадає в очі, це ціла купа смуг прокруток, які розташовуються де потрібно і не потрібно. Для того, щоб їх видалити, користуємось допомогою атрибуту *scrolling* – тега `<frame>`. Він може мати одне з трьох значень:

- no - ніколи не показувати смугу прокрутки;
- yes - завжди показувати;
- auto - показувати у тому разі якщо вона потрібна.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800*">
<frame src="dekor.html"scrolling="no">
<frameset rows="120*">
<frame src="logotype.html"scrolling="no">
<frameset cols="200, 600">
<frame src="menu.html">
<frame src="text.html">
```

```
</frameset>
</frameset>
<frame src="dekor.html"scrolling="no">
</frameset>
</html>
```

Поля фреймів (відстань від меж фрейма до тексту або картинки) задаються в пікселях за допомогою атрибутів *marginwidth* і *marginheight* тега `<frame>`.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800*">
<frame src="dekor.html" scrolling="no">
<frameset rows="120*">
<frame src="logotype.html" scrolling="no"marginwidth="0"
marginheight="0">
<frameset cols="200, 600">
<frame src="menu.html">
<frame src="text.html"marginwidth="10"
marginheight="10">
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no">
</frameset>
</html>
```

Рамки навколо фреймів

В останньому прикладі навівши курсор на межу фрейма, курсор набуває іншого вигляду (захопити і перемістити). Для того, щоб заборонити користувачеві гратися з розмірами вікон, для тега `<frame>` придумали атрибут *noresize*.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800*">
<frame src="dekor.html" scrolling="no" noresize>
<frameset rows="120*">
<frame          src="logotype.html"          scrolling="no"
marginwidth="0" marginheight="0" noresize>
<frameset cols="200, 600">
<frame src="menu.html" noresize>
<frame          src="text.html"          marginwidth="10"
marginheight="10" noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>
```

Атрибут *border* задає ширину в пікселях цих самих рамок між фреймами. Пишеться він всередині тега *<frameset>*. Як і раніше, значення *border="0"* зовсім позбавить нас від рамок.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800,*" border="0">
<frame src="dekor.html" scrolling="no" noresize>
<frameset rows="120,*" border="0">
```

```
<frame src="logotype.html" scrolling="no"
marginwidth="0" marginheight="0" noresize>
<frameset cols="200, 600"border="0">
<frame src="menu.html" noresize>
<frame src="text.html" marginwidth="10"
marginheight="10" noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>
```

Фрейми і посилання

Фрейму, в якому ми хотіли б відкривати які-небудь документи, необхідно присвоїти індивідуальне ім'я.

Пишеться це так:

```
<frame src="text.html"name="osnovnoe"> ім'я можна
придумати будь-яке, головне його не забути.
```

А тепер в документі з посиланнями(в даному випадку це файл menu.html) необхідно вказати мету, тобто визначити в якому вікні варто відкривати той або інший документ.

Пишеться так:

```
<a href="text 1.html"target="osnovnoe">Боби в горщику
по-італійськи</a>.
```

Приклад:

Файл index.html

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800,*" border="0">
<frame src="dekor.html" scrolling="no" noresize>
<frameset rows="120,*" border="0">
```

```
<frame src="logotype.html" scrolling="no"
marginwidth="0" marginheight="0" noresize>
<frameset cols="200, 600" border="0">
<frame src="menu.html" noresize>
<frame src="text.html" name="osnovnoe" marginwidth="10"
marginheight="10" noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>
```

Файл menu.html

```
<html>
<head>
<title>фрейму</title>
</head>
<body bgcolor="#e8e8e8" background="fon 1.jpg"
link="#b40000" alink="#900000" vlink="#900000">
<font face="Monotype Corsiva" size="6"
color="#ff0000">Меню:</font><hr>
<font face="Comic Sans MS" size="3">
<a href="text.html" target="osnovnoe">Сун 3
шампінйонів</a><hr>
<a href="text 1.html" target="osnovnoe">Боби в горицку
по-італійськи</a><hr>
<a href="text 2.html" target="osnovnoe">Австралійський
літній салат</a><hr>
... ..
</font>
</body>
</html>
```

Також документ можна відкрити в окремому вікні.

Пишеться так:

`<a href="text 1.html"target="_blank">Боби в горицику по-італійськи</a>.`

Або ж, присвоївши атрибуту *target* значення *_top*, відкрити його в цьому ж вікні браузера, але на весь екран, "обнуливши" при цьому все, що б там не знаходилося. пишеться так:

`<a href="text 1.html"target="_top">Боби в горицику по-італійськи</a>.`

Плаваючий фрейм

Іноді в сторінку необхідно в окремому вікні вставити інший HTML документ або навіть ряд таких документів. Для виконання цієї мети існує тег `<iframe>` - так званий плаваючий фрейм.

Цей тег має ряд атрибутів :

src - обов'язковий атрибут, що вказує шлях до сторінки, що відкривається;

width - ширина плаваючого фрейма в пікселях або відсотках;

height - висота плаваючого фрейма.

scrolling - показ смуги прокрутки

no - ніколи не показувати смугу прокрутки

yes - завжди показувати

auto - показувати у тому разі якщо вона потрібна

align - вирівнювання плаваючого фрейма

left - ліворуч

right - справа

top - вище

bottom - нижче

frameborder - наявність рамки навколо плаваючого фрейма

1 - включити рамку

0- вимкнути рамку

Все разом пишеться так:

```
<iframe src="primer.html" width="300" height="250" align="left" scrolling="auto" frameborder="1" ></iframe>
```

Приклад документу з плаваючим фреймом:

```
<html>
<head>
<title>Плаваючий фрейм</title>
</head>
<body>
<center><h2>Плаваючий фрейм</h2></center>
У цю сторінку введений так званий "плаваючий фрейм".
У окремому вікні він відкриває для показу інший документ
html.
<br>
<iframe src="primer.html" width="250" height="250" align="left" scrolling="auto" frameborder="1"></iframe>
<br>
... ..
<br>
</body>
</html>
```

Noframes

Деякі браузери не підтримують фреймову структуру документу або неправильно її інтерпретують. Крім того часто користувачі в налаштуваннях своїх браузерів умисно відключають підтримку фреймової структури html документу.

Для того що б дати зрозуміти користувачеві, що його браузер/налаштування браузеру не підтримують фрейми існує тег `<noframes>`.

Тег `<noframes>` виводить текст, поміщений в нього, у тому випадку, якщо браузер користувача не підтримує фрейми або вони примусово вимкнені в його налаштуваннях. Якщо ж

фрейми підтримуються браузером користувача, то цей тег просто ігнорується.

Приклад:

```
<html>
<head>
<title>фрейми</title>
</head>
<frameset cols="*,800,*" border="0">
<noframes>Даруйте, але Ваш браузер не підтримує
фрейми.</noframes>
<frame src="dekor.html" scrolling="no" noresize>
<frameset rows="120,*" border="0">
<frame          src="logotype.html"          scrolling="no"
marginwidth="0" marginheight="0" noresize>
<frameset cols="200, 600" border="0">
<frame src="menu.html" noresize>
<frame          src="text.html"          marginwidth="10"
marginheight="10" noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>
```

Результат прикладу буде помітний, якщо браузер дійсно не підтримує фрейми.

Тег `<noframes>` має бути розташований усередині тега `<frameset>`.

З плаваючим фреймом все простіше, достатньо написати потрібний текст `<iframe></iframe>` і цей напис буде виводиться на екран, у тому разі якщо браузер не підтримує фрейми.

Ось так:

```
<iframe src="primer.html" width="300" height="250" align="left" scrolling="auto" frameborder="1">Даруйте, але Ваш браузер не підтримує фрейми.</iframe>
```

Корисні поради:

Перш ніж почати створювати сторінку з використанням фреймової структури, проаналізуйте її макет, розміри кожного вікна, наявність або відсутність в них смуг прокруток і т. д. Після цього можна створювати HTML файли, що підключаються.

Використовуйте тег `<noframes>`. Пам'ятайте, що якщо у вашому браузері сайт працює і відображає все те, що Ви замислювали, це не означає, що і в інших користувачів все буде так само гарно відображатись.

Рекомендована література:

1. Інформатика (профільний рівень) : підруч. для 10 кл. закл. загал. серед. освіти /В. Д. Руденко, Н. В. Речич, В. О. Потієнко. — Харків : Вид-во «Ранок», 2018.
2. Інформатика : підруч. для 10(11) кл. закладів загальної середньої освіти / Н. В. Морзе, О. В. Барна. — К. : УОВЦ «Оріон», 2018.
3. Інформатика (рівень стандарту): підручник для 10 (11) класу закладів загальної середньої освіти / Й.Я.Ривкінд, Т. І.Лисенко, Л.А. Чернікова , В.В. Шакотько. — Харків: Вид-во «Ранок», 2018.
4. Інформатика та програмування. Частина 1 : навчальний посібник /Юрченко І.В.. — Чернівці: Вид-во «Книги–ХХІ», 2011.

ПРО УКЛАДАЧІВ

Холупняк Катерина Олександрівна – з 2011 року викладач Черкаського державного бізнес-коледжу. Закінчила Черкаський державний бізнес-коледж за спеціальністю „Обслуговування комп’ютерних систем та мереж” (2006 р.), Черкаський державний технологічний університет за спеціальністю „Комп’ютерні мережі ” (2010 р.), Черкаський національний університет ім. Б. Хмельницького за спеціальністю „Управління навчальним закладом” (2015 р.). Спеціаліст першої категорії.

Ковальська Наталія Володимирівна – голова циклової комісії комп’ютерних систем та мереж Черкаського державного бізнес-коледжу. Закінчила Черкаський державний університет ім. Б. Хмельницького за спеціальністю „Математика та інформатика” (2001 р.). З 2002 року викладач Черкаського державного бізнес-коледжу. Спеціаліст вищої категорії, старший викладач. Є автором методичних розробок: “Інформатика та комп’ютерна техніка. Курс лекцій” (2005р.), “Основи інформатики. Курс лекцій та тестові завдання ” (2011 р.), “Інформатика. Система управління базами даних MS Access. Частина 1” (2018р.), Основи інформаційних технологій. Робота в середовищі MS Excel. Збірник завдань (2019).

Навчальне видання

Холупняк Катерина Олександрівна
Ковальська Наталія Володимирівна

**Основи інформаційних технологій.
HTML для початківців**

Редактор Н.А. Ковальська
Коректор Дражан Н. С.
Комп'ютерний набір
Холупняк К.О., Ковальська Н.В.

Підписано до друку _____ р. Формат 60х84^{1/16}
Папірофсет. Гарнітура Times New Roman. Друк офсетний.
Умов. друк. арк. 1,34 Тираж 30 прим. Зам. № 262

За довідками з питань реалізації
звертатись за тел. (0472) 64-05-15