Міністерство аграрної політики та продовольства України
Вінницький національний аграрний університет
ННІ аграрної економіки

Кафедра економічної кібернетики

WEB-ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

для виконання лабораторних та самостійних робіт бакалаврами напряму 6.050100– «Економічна кібернетика»

Вінниця 2011

Web-програмування. Методичні вказівки для виконання лабораторних та самостійних робіт бакалаврами напряму 6.050100– «Економічна кібернетика» / Паламарчук Є.А., Яцковська Р.О. - Вінниця: ВНАУ, 2011.- 77с.

Укладачі: Паламарчук Є.А., доцент, к.т.н.
Яцковська Р.О., асистент

Рецензенти: Кравченко Ю.С., к.ф.-м.н., доцент кафедри електроніки ВНТУ

Поліщук Н.В. к.е.н., доцент кафедри економічної кібернетики ВНАУ

Коротка анотація

Методичні вказівки для виконання лабораторних робіт з дисципліни "Web-програмування" містять структурно-модульну схему курсу, розбиття дисципліни на модулі та розподіл балів за категоріями діяльності студента, 6 лабораторних робіт, вимоги до виконання та оформлення лабораторних робіт та список рекомендованої літератури.

Призначені для використання бакалаврами напряму 6.050100– «Економічна кібернетика».

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

Рекомендовано науково-методичною радою Вінницького національного аграрного університету протокол № __ від „___” _____ 2011 р.

ЗМІСТ

ВСТУП

Метою вивчення дисципліни є формування сучасного рівня інформаційної та комп'ютерної культури, набуття практичних навичок роботи при створенні інтерактивних технологій..

Основне завдання навчальної дисципліни полягає у вивченні таблиці каскадних стилів, створення інтерактивних Веб-сайтів з використанням мови JavaScript і динамічного HTML, набутті навичок програмування на JavaScript, перевірки даних користувача на стороні клієнта. вивчення основи програмування на PHP, збереження та отримання даних..

Предметом дисципліни є використання баз даних при розробці Веб-застосувань. Проектування веб-баз даних. Створення баз даних. З'єднання з сервером MySQL засобами PHP. Виконання запитів і обробка результатів

У результаті вивчення дисципліни студенти повинні:

- знати специфікації CSS, інтеграцію таблиці стилів в HTML документ, типи селекторів і особливості їх використання, наслідування стилів, стилі оформлення тексту, блочна розмітка веб-сторінки;
- мати уявлення про створення інтерактивних Веб-сайтів з використанням мови JavaScript і динамічного HTML;
- мати уявлення про взаємодію з сервером за технологією Ajax, використання плагінів;
- набути навички роботи з програмування на PHP. Збереження та отримання даних. Використання масивів. Робота з текстом. Регулярні вирази. Повторне використання коду і створення функцій. Об'єктно-орієнтоване програмування на PHP. Взаємодія з файловою системою і сервером. Робота з датою і часом. Створення графіки. Керування сесіями;
- набути навички використання використання баз даних при розробці Веб-застосувань. Проектування веб-баз даних. Створення баз даних. З'єднання з сервером MySQL засобами PHP. Виконання запитів і обробка результатів.;

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

Паламарчук Євген Анатолійович,
Яцковська Римма Олександрівна,

WEB-ПРОГРАМУВАННЯ

Набір і редагування авторські

Технічний редактор *Олександр Романов*

Верстка

Підписано до друку Формат 60x84/16.
Папір офсетний. Друк різографічний.
Тираж прим.

Віддруковано у редакційно-видавничому відділі
Вінницького державного аграрного університету
21008, м. Вінниця, вул. Сонячна, 3

Вивчення дисципліни:

- надає підґрунтя для подальшого засвоєння можливостей використання комп'ютерної техніки у спеціальних дисциплінах навчального плану студентів економічних спеціальностей всіх форм навчання (використання інформаційних систем в різних галузях господарства, економетрія, ризикологія тощо);
- формує інформаційну культуру, що підвищує загальну компетенцію майбутніх фахівців з економіки та менеджменту, сприяє високій конкурентоспроможності випускників на українському і європейському ринку праці та є основою їх висококваліфікованої професійної діяльності.

Програма курсу передбачає навчання в формі лекцій і лабораторних робіт. Для практичного засвоєння основних тем дисципліни лабораторні роботи проводяться з застосуванням персональних комп'ютерів, локальних мереж та мережі Intranet в комп'ютерних класах ВНАУ.

Лабораторна робота передбачає самостійне виконання кожним студентом комплексного індивідуального завдання фахового спрямування та має за мету систематизацію знань та продовження формування відповідних навичок кожного студента.

Структура орієнтовного розподілу балів за модулями навчальної дисципліни, що присвоюється студентам

Вид контролю	Модуль	Тема	Навч. заняття (підготовка до виконання)	Виконання індивід. завдань (ОР, реферат, РГР, РР та ін.)	Модульний (змістово-модульний контроль)	Всього балів (сума 4+5+6)
1	2	3	4	5	6	7
Поточний контроль	1 Використання PHP-технологій	Лекції: 1. Введення до курсу. “CSS”. 2. Базові властивості CSS. 3. Управління властивостями шрифтів. 4. Стилі списків. 5. Способи позиціонування елементів на сторінці або відносно інших об’єктів. Властивості таблиць. 6. Передача параметрів між Web-стоінками. Способи підтримання ідентифікації користувача. 7. Перегляд Cookie	**8**			**8**
		Лаб. роботи: 1. Дослідження каскадних стилів CSS. Частина 1. 2. Дослідження каскадних стилів CSS. Частина 2. 3. Дослідження методів передачі даних між WEB-сторінками GET, POST, SESSIONS.	**5**	**4**	**5**	**14**
		СРС: 1. Використання php-операторів 2. Технологія створення таблиць. 3. Основи роботи з використанням web-технологій		**3**		**3**
		Зах. мод. 1			**10**	**10**
	Всього за модуль 1					**35**

ДЖЕРЕЛА В INTERNET

1. http://www.tnpu.edu.ua/kurs/413/?page=lec4.php
2. http://programer.org.ua/index.php?mainpart=2
3. http://www.znannya.org/
4. http://uk.wikipedia.org
5. www.diasoft.ru
6. www.dtkr.ua
7. www.knigka.org.ua/category/office_applications
8. www.programbank.ru
9. www.softlab.ru
10. www.vsau.vin.ua
11. www.williamspublishing.com

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Максим Кузнєцов, Ігор Симдянов, Сергій Голишев. PHP 5. Практика створення Web-сайтів**:** БХВ-Петербург, - 2005р., - 948 с.

2. Джордж Шлосснейгл. Професійне програмування на PHP**:** Вільямс, 2006 р., - 624 с.

3. Люк Веллинг, Лора Томсон. Розробка Web-додатків за допомогою PHP і MySQL: Вільямс, 2007р., - 880 с.

4. Зольників Д. З. PHP 5**:** НТ Пресс, 2007 р., 256 с.

5. Ларри Ульман. Основи програмування на PHP: ДМК прес, 2001 р., - 288 с.

6. Баженов В. А., Венгерським П. С. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. К.: Каравела, 2007. –640 с.

7. Глушков С.В., Сурядний А.С. Персональний компъютер. Харьков. ФОЛИО, - 2004 г., - 282 с.

8. Дибкова Л. М. "Інформатика та комп'ютерна техніка", К., "Академія", 2002р. - 325 с.

9. Економічна інформатика Практикум: Навчальний посібник/ С.Д. Мамченко, В.А. Одинець. - К.: Знання, 2008. - 710с.

10. Економічна інформатика: Навчальний посібник/ В.С. Григорків, Л.Л. Маханець, Р.Р. Білоскурський и др.; Чернівці: Книги-ХХІ, 2008. - 464с.

11. Інформатика та комп'ютерна техніка: Навчальний посібник/ М.В. Макарова, Г.В. Карнаухова, С.В. Запара; Ред. М.В. Макарова. - Суми: Університетська книга, 2008. - 665с.

12. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: підручник/ В.А. Баженов, П.С. Венгерський, В.М. Горлач, І.М. Дудзяний. - К.: Каравела, 2008. - 639 с.

13. Олександр Качанов. Буквар по PHP і MySQL, 2000 р, 27 с.

14. Энди Харрис. PHP/MySQL для початківців: Кудиц-образ, 2005р., - 284 с.

15. К Косентіно. PHP web професіоналам: Посібник. К.: Видавнича група BHV, 2001. – 250 с.

Продовження таблиці

1	2	3	4	5	6	7
Поточний контроль	2 Робота з РБД	Лекції 1. Sessions. 2. Створення спливаючих вікон за допомогою Java-скрипт і їх використання у у Web-дизайнах і проектах. 3. Застосування AJAX-технологіїпри передачі даних від HTMLсторінки до сервера БД. 4. Web-дизайн. 5. Аутентифікація. Проблеми захисту паролів. 6. Електронна пошта 7. Системи команд управління IMAP-сервера.	**8**			**8**
		Лаб. роб. 1. Дослідження COOKIE. 2. . Засоби безпеки у WEB-програмах. Хеш-функції. 3. Дослідження взаємодії WEB-проектів із SMTP-серверами при надсиланні електронної пошти.	**5**	**4**	**5**	**14**
		СРС 1. JavaScript. AJAX та PHP-технології. 2. Електронна пошта 3. Web-дизайн		**3**		**3**
		Зах. мод. 2			**10**	**10**
	Всього за модуль 2					**35**
Всього за поточний контроль*						**70**
Підсумковий контроль (екзамен)						**30**
Разом						**100**

ЛАБОРАТОРНА РОБОТА №1

Тема: Дослідження каскадних стилів CSS. Частина 1.

Мета: Набуття навичок створенні і застосуванні каскадних стилів CSS. Базові властивості.

Теоретичні відомості

CSS (cascading style sheets – каскадні таблиці стилів) – одна з базових WEB-технологій сучасного інтернету. CSS-код – це список інструкцій для браузера про те, – *яким чином* відтворювати елементи веб-сторінок. Інструкції CSS зберігаються або в окремому файлі, що має розширення *.css, або у вигляді спеціального блоку команд в тексті WEB-сторінки або безпосередньо в тегах. Под «елементами» маються на увазі теги XHTML/HTML і їх вміст.

Головна ідея CSS полягає у тому, щоби код HTML не вміщував елементи оформлення типу встановлення кольору, гарнітури кегля і т. і. В ідеальному варіанті WEB-сторінка повинна вміщувати лише теги логічного форматування, а вид елементів задаватись через стилі. При подібному розділенні формування дизайну і верстка сайту можуть вестись паралельно.

Під способами застосування CSS розумітимемо форму декларування стилю на HTML-сторінці і форму зв'язування опису стилю відображення елемента розмітки із самим елементом, тобто у якій формі автор сторінки (або дизайнер) описує стиль, і як і в якій формі його використовує.

Отже, розрізняють чотири способи застосування стилів:

- **вбудовування** у теги документа – дозволяє змінити форматування конкретних елементів сторінки;
- **впровадження** – дозволяє задавати всі правила таблиці стилів в самому документі;
- **зв'язування** – дозволяє використовувати одну таблицю стилів для форматування багатьох сторінок;
- **імпортування** – дозволює вбудовувати у документ таблицю стилів, розміщену на сервері.

Важливо відзначити, що в усіх браузерах, крім Internet Explorer імпорт стилю не підтримується.

41. Кодування даних в електронній пошті. php-функція кодування та приклад її використання у формуванні електронного листа.
42. Структура електронного листа. Кодовані і некодовані поля.
43. Поштові сервери POP та POP3. Призначення. Принцип роботи. Основні команди управління. Аутентифікація.
44. Поштові IMAP-сервери. Призначення. Принцип роботи. Основні команди управління. Аутентифікація.
45. Порівняльна характеристика поштових серверів типу POP та IMAP. Переваги і недоліки.
46. Протокол роботи з IMAP-сервером. Команди обробки пошти та управління поштовими скринями.
47. Використання команд php-бібліотеки IMAP у створенні простого поштового скрипту.
48. Використання Javascript, Ajax та Jquery у інтерактивних WEB-сторінках. Принцип роботи.

23. Переваги і недоліки методу SESSION. Принцип роботи. Тривалість. Безпека. Приклади його застосування.

24. Принцип безпечності методу SESSION.

25. Використання механізму COOKIE у WEB-додатках. Основні PHP-оператори. Принцип роботи. Тривалість. Безпека. Приклади його застосування.

26. Побудова COOKIE. Поля та їх призначення.

27. Спільні характеристики і механізм дії у методах COOKIE і SESSIONS?

28. Операції з COOKIE. Встановлення, зміна та читання у WEB-додатках. Рознесення операцій між WEB-додатками.

29. Часові характеристики COOKIE. Використання їх у WEB-додатках.

30. Використання COOKIE для збереження персональних даних. Приклади використання.

31. Обмін інформацією між WEB-сторінками за допомогою COOKIE? Приклади використання.

32. Проблеми безпеки WEB-додатків при використанні COOKIE.

33. Проблеми безпеки WEB-додатків. Передача інформації і збереження паролів.

34. Хеш-функції. Призначення і основні властивості. Що означає незворотність перетворення хеш-функції?

35. Стійкість хеш-функцій. Колізії. Лавинний ефект.

36. Стійкість хешів. Методи **brute force** і **rainbow**. Вимоги до структури паролів при створенні стійких хешів.

37. Методи формування стійких хешів. Метод **salt.**

38. Методи передачі електронної пошти. Поштові SMTP-сервери. Призначення та основні функції. Аутентифікація.

39. Використання сокетів у передачі даних. Основні параметри з'єднання та php-команди для організації обміну інформацією між WEB-додатком і сервером.

40. Використання портів у сокетах. Відкриті та закриті протоколи обміну даними. Основні параметри з'єднання та php-команди для організації обміну інформацією між WEB-додатком і сервером.

Вбудовування

```
<H1 STYLE="font-weight:normal; font-style:italic; font-size:10pt;">
Заголовок першого рівня
</H1>
```

Атрибут style можна застосувати всередині будь-якого елемента розмітки. Наприклад, ми можемо через style визначити ширину і вирівнювання елемента hr (горизонтальна лінія):

```
<HR STYLE="width:100px;">
```

Впровадження

Застосування елемента STYLE – це основний спосіб впровадження каскадних таблиць стилів у HTML-документ. Крім керування відображенням елементів розмітки, елемент STYLE дозволяє описувати стильові властивості елементів, які можна змінювати при програмуванні на JavaScript.

Елемент STYLE дає можливість визначити стиль відображення для:

- стандартних елементів HTML-розмітки;
- довільних класів (селектор CLASS);
- HTML-об'єктів (селектор ID).

Стандартні елементи розмітки описуються в елементі STYLE наступним чином:

```
<HEAD>
<STYLE>
   p { color:darkred; text-align:justify; font-size:8pt; }
</STYLE>
</HEAD>
   <BODY>
   ...
   <P>Цей параграф ми використовуємо як приклад
                застосування опису стилю для стандартного
                елемента HTML-розмітки.
   </P>
...
   </BODY>
```

Тепер усі параграфи документа будуть відображатися стилем з елемента STYLE, якщо тільки стиль не буде яким-небудь способом перевизначений. У STYLE можна визначити стиль будь-якого елемента розмітки.

Зв'язування

Посилання на опис стилю, розташованого за межами документа, здійснюється за допомогою елемента LINK, що розміщають в елементі HEAD. Зовнішнім описом може бути файл, що містить опис стилів. Опис стилів у цьому файлі буде по синтаксису в точності збігатися зі змістом елемента STYLE.

Нижче наведений приклад посилання на зовнішній опис стилів:

```
<LINK TYPE="text/css" REL="stylesheet" HREF="my_css.css">
```

Тут важливі значення атрибутів REL і TYPE. Атрибут REL повинен мати значення stylesheet. Type може приймати значення text/css або text/javascript. Атрибут HREF задає універсальний показчик ресурсу (URL) для зовнішнього файлу опису стилів. Це може бути посилання на файл із будь-яким ім'ям, а не лише на файл із розширенням *.css.

Імпортування

Імпорт опису стилів у дечому нагадує вказівку на зовнішній опис стилю. Імпортувати стиль можна або всередину елемента STYLE, або всередину зовнішнього файлу, що являє собою опис стилю.

Оператор імпорту стилю повинен передувати всім іншим описам стилів:

```
<STYLE>
@import:url(http://intuit.ru/style.css)
A { color:cyan;text-decoration:underline; }
</STYLE>
```

Імпортований стиль можна перевизначити або через опис елемента в STYLE, або через атрибут елемента STYLE.

Синтаксис

Формально стиль відображення елементів розмітки задається посиланням в елементі розмітки на селектор стилю. Синтаксис опису стилів у загальному вигляді можна подати в такий спосіб:

```
selector[, selector[, ...]]
     { attribute:value;
     [attribute:value;...] }
```


ПРОГРАМНІ ПИТАННЯ

1. Каскадні таблиці стилів CSS. Принципи побудови. Керування дизайном сайту.
2. Основні властивості каскадних таблиць стилів CSS.
3. Інструкції CSS та їх розташування?
4. Призначення селекторів CSS. Приклади використання селекторів.
5. Класи у селекторах CSS. Приклади використання селекторів класів.
6. ID-селектори CSS? Приклади використання ID-селекторів.
7. Застосування стилів CSS у декількох селекторах.
8. CSS-селектори контексту. Приклади використання.
9. CSS. Селектори нащадків? Приклади використання.
10. Базові властивості тексту і управління ними у CSS.
11. Базові властивості фону і управління ними у CSS.
12. Базові властивості шрифтів і управління ними у CSS.
13. Базові властивості рамок і управління ними у CSS.
14. Використання шарів у WEB-дизайні. Властивості шарів і управління ними за допомогою CSS?
15. Базові властивості гіперпосилань і управління ними у CSS.
16. Базові властивості таблиць і управління ними у CSS.
17. Базові властивості списків і управління ними у CSS.
18. У чому різниця методів GET і POST?
19. Механізми передачі даних між WEB-додатками. Їх порівняльна характеристика. Використання у формах, гіперпосиланнях і WEB-сторінках.
20. Принцип обміну даних WEB-сервера з браузерами, який забезпечує можливість одночасної і незалежної їх роботи з WEB-додатками з декількох комп'ютерів, що мають одну IP-адресу.
21. Переваги і недоліки методу GET. Принцип роботи. Тривалість. Безпека. Приклади його застосування.
22. Переваги і недоліки методу POST. Принцип роботи. Тривалість. Безпека. Приклади його застосування.

5. Розкоментуйте останні два рядки і опишіть відповіді SMTP-сервера на кожну команду вашого поштового скрипту. Результати занотуйте в блог.

Контрольні запитання:

1. Що таке SMTP-сервер і як розшифровується абревіатура?
2. Яка послідовність діалогу з SMTP-сервером при надсиланні електронної пошти?
3. Чому в SMTP-серверах використовується авторизація?
4. Що таке сокет і як він використовується у поштовому скрипті?
5. Які php-команди відкривають і закривають сокет?
6. Що потрібно мати, щоби відкрити сокет?
7. Для чого використовується порт в SMTP-протоколі?
8. Для чого використовується **ssl** і **tls** протоколи в поштових серверах? Що буде відбуватись із даними, якщо їх вимкнути?
9. Для чого і чому кодуються дані в електронній пошті? Яка функція відповідає у скрипті за їх кодування?
10. Яка php-команда у скрипті відповідає за надсилання даних через сокет? Що вказується у якості її параметрів?
11. Яка php-команда у скрипті відповідає за одержання даних через сокет? Що вказується у якості її параметрів?
12. Для чого в скрипті використовується асоціативний масив **$log**?
13. Для чого в протоколі використовується змінна **$CR**?
14. Покажіть місце у скрипті, де поштовому серверу надсилається ознака кінця повідомлення? Що вона собою представляє?

або

```
selector selector [selector ...]
      { attribute:value;
      [attribute:value;...] }
```

У першому варіанті перераховані селектори, для яких діє даний опис стилю. Другий варіант задає ієрархію вкладеності селекторів, для сукупності яких визначений стиль.

Як селектор можна використовувати ім'я елемента розмітки, ім'я класу й ідентифікатор об'єкта на HTML-сторінці.

Атрибут (attribute) визначає властивість відображуваного елемента, наприклад лівий відступ параграфа (margin-left), а значення (value) — значення цього атрибута, наприклад, 10 типографських пунктів (10 pt).

Селектор — ім'я елемента розмітки

Якщо треба визначити загальний стиль усіх сторінок, то прописуються стилі для всіх елементів HTML-розмітки, що будуть використовуватися на сторінках.

Такий спосіб створення сайту дозволяє змінювати зовнішній вигляд усіх сторінок шляхом внесення змін у файл опису стилів, а не у файли HTML-сторінок.

Зовнішній файл при цьому може виглядати наступним чином:

```
I, EM {color:#003366;font-style:normal}
A I {font-style:normal;font-weight:bold;text-decoration:line-through}
```

У першому рядку цього опису перераховані селектори-елементи, що будуть відображатися однаково:

```
<I>Це курсив</I> і це також <EM>курсив</EM>
Останній рядок визначає стиль відображення вкладеного
в гіпертекстове посилання курсиву:
<A NAME=empty><I>гіперпосилання курсивом</I></A>
```

У даному випадку перевизначення полягає в тому, що текст відображається всередині гіпертекстового посилання перекресленим, причому жирним шрифтом.

Селектор - ім'я класу

Ім'я класу визначає опис класу елементів розмітки, що будуть відображатися однаково. Для того, щоб віднести елемент

розмітки до того або іншого класу, потрібно скористатися його атрибутом CLASS:

```
<STYLE>
.test {color:white;background-color:black;}
</STYLE>
...
<P CLASS="test">
    Цей параграф ми відобразимо білим кольором по чорному тлу
</P>
...
<P>
    Це <A CLASS="test"> гіпертекстове посилання </A>
    ми відобразимо білим кольором по чорному тлу.
</P>
```

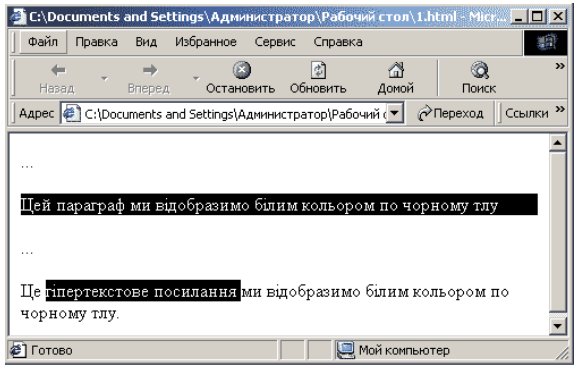


Рис. 1.1 Селектор - ім'я класу

Початкову крапку в імені класу можна опустити. Вона задається з розумінь збереження єдності опису. Наприклад, можна визначити класи відображення однотипних елементів розмітки:

```
a.menu { color:red;background-color:white;text-decoration:none; }
a.paragraph { color:navy;text-decoration:underline; }
```

У даному прикладі клас гіпертекстових посилань menu має один опис стилю, а клас гіпертекстових посилань paragraph – зовсім інше. При цьому кожен з цих класів не можна застосувати до інших елементів розмітки, наприклад, параграфові або спискові. Якщо ім'я елемента розмітки не задано, це означає, що клас можна застосувати до будь-якого елемента розмітки.


```
//надсилаємо логін користувача $username
fputs($sc, base64_encode($username) . $CR);
$sr = fgets($sc, 4096);
$log['authusername'] = "$sr";

//надсилаємо пароль $password
fputs($sc, base64_encode($password) . $CR);
$sr = fgets($sc, 4096);
$log['authpassword'] = "$sr";

//а тепер надсилаємо команду MAIL FROM: - від кого $from
fputs($sc, "MAIL FROM: <$from>" . $CR);
$sr = fgets($sc, 4096);
$log['mailfromresponse'] = "$sr";

//і кому надсилається лист. Команда RCPT TO: - $to
fputs($sc, "RCPT TO: <$to>" . $CR);
$sr = fgets($sc, 4096);
$log['mailtoresponse'] = "$sr";

//ну і тепер за командою DATA власне далі розпочнеться і сам лист
fputs($sc, "DATA" . $CR);
$sr = fgets($sc, 4096);
$log['data1response'] = "$sr";

//спочатку у листі зробимо заголовки
$headers = "MIME-Version: 1.0" . $CR;
$headers .= "Content-type: text/html; charset=utf-8" . $CR;
$headers .= "To: $nameto <$to>" . $CR;
$headers .= "From: $namefrom <$from>" . $CR;
//В тепер надсилаємо це все разом із самим повідомленням $message
//В кінці ОБОВ`ЯЗКОВО треба надіслати крапку: це команда, що означає кінець
листа
fputs($sc, "To: $to\r\nFrom: $from\r\nSubject:
$subject\r\n$headers\r\n\r\n$message\r\n.\r\n");
$sr = fgets($sc, 4096);
$log['data2response'] = "$sr";
// роз'єднуємось... Команда QUIT
fputs($sc,"QUIT" . $CR);
$sr = fgets($sc, 4096);
$log['quitresponse'] = "$sr";
$log['quitcode'] = substr($sr,0,3);
fclose($sc);
// протокол роботи з smtp-сервером повеpемо до викликаючої програми
return($log);
}

// приклад використання функції
$log = MySendMail('stud@i.ua', 'Микола Куринюк', 'teacher@mail.ua', 'Антоніна
Стець', 'Вітання з днем вчителя', '.....');
//echo "Виведемо всі діагностичні повідомлення при надсилання пошти:<hr>";
//foreach($log as $k => $v)    echo "a['$k']=$v <br>";

?>
```

```
<html><head><meta content='text/html; charset=UTF-8' http-equiv='content-
type'></head><body>

<?php
function MySendMail($from, $namefrom, $to, $nameto, $subject, $message)
{
// функція для надсилання листів через з'єднання socket до SMTP-сервера
// -------------------------------------------------------------------
// тут треба прописати необхідні атрибути з'єднання і аутентифікації
$smtpServer = "tls://smtp.gmail.com"; // приєднання за криптованим протоколом
tls

//$smtpServer = "smtp.svitonline.com"; //відкрите з'єднання
//$port = "25"; // можливий порт 465 або 587

$port = "465"; // можливий порт 587 або 465
$timeout = "45"; //таймаут. 45 сек для повільних з'єднань
$username = "abc@gmail.com"; //акаунт користувача на поштовому сеервері
$password = "%r^trffds"; //пароль користувача до поштового сервера
$localhost = $_SERVER['REMOTE_ADDR']; //одержуємо реальну IP-адресу
нашого комп'ютера
$CR = "\r\n"; //роздільник нового рядку — пара спецсимволів 0d 0a у
шістнадцядковому форматі
//-------------------------------------------------------------------------------------------

// $sr = $smtpResponce - відповідь smtp-сервера
// $sc = $smtpConnect — канал з'єднання до smtp-сервера
// $log - масив для діагностики процесу з'єднання і надсилання пошти
//приєднуємось до поштового сервера із вказаним портом через сокет

$sc = fsockopen($smtpServer, $port, $errno, $errstr, $timeout);
$sr = fgets($sc, 4096); // читаємо блок з 4096 байтів
if(empty($sc))
{
        $output = "Помилка з'єднання: $sr";
        echo $output;
        return $output;
}
else
{
        $log['connection'] = "Приєднались до: $sr";
}

//надсилаємо команду HELO знову після встановлення TLS-з'єднання
  fputs($sc, "HELO $localhost". $CR);
  $sr = fgets($sc, 4096); // читаємо блок з 4096 байтів
  $log['heloresponse2'] = "$sr";

//надсилаємо команду AUTH LOGIN для авторизації логіну
fputs($sc,"AUTH LOGIN" . $CR);
$sr = fgets($sc, 4096);
$log['authrequest'] = "$sr";
```


Селектор - ідентифікатор об'єкта

Об'єктна модель документа (Document Object Model) описує документ як дерево об'єктів. Об'єктами є: сам документ, його розділи (елемент DIV), зображення, параграфи, додатки і т.п. Кожному з цих об'єктів можна дати ім’я і звертатися до нього по імені.

Застосування ідентифікатора об'єкта виправдано ще й у випадку модифікації атрибута опису стилю для даного об'єкта в його CSS-описі. Замість двох описів класів, що відрізняються лише одним з параметрів, можна створити один опис класу й опис ідентифікатора об'єкта. Опис стилю для об'єкта задається рядком, у якому селектор являє собою ім'я цього об'єкта з передуючим символом "#":

```
a.mainlink { color:darkred; text-decoration:underline;font-style:italic; }
#blue { color:#003366 }
...
<A CLASS=mainlink>основна гіпертекстова посилання</A>
<A CLASS=mainlink ID=blue>модифікована гіпертекстове посилання</A>
```

Існує ще атрибут name в елемента розмітки. При ідентифікації об'єкта Netscape Navigator зазвичай має справу саме з цим атрибутом, а Internet Explorer — з атрибутом ID.

Наслідування і перевизначення

По-перше, існує ієрархія елементів розмітки. По-друге, властивості цих об'єктів можуть наслідуватися. У такий спосіб у дереві об'єктів утвориться гілка, що веде до елемента дерева – елемента розмітки, наприклад, елемента списку або параграфа. Його властивості визначаються елементами розмітки, в які вкладений елемент, і описом стилю для даного елемента.

Текст на зображенні (Р и с . 1.2) закодований у термінах розділів і списків.

У такий спосіб відступи відраховуються щодо елемента, в який вкладений поточний елемент. Усі параметри, що не були перевизначені в поточному елементі, наслідуються зі старшого по ієрархії елемента. Останнє добре продемонстровано в застосуванні стилів відображення списку, що вкладений у розділ і тому відображається курсивом.

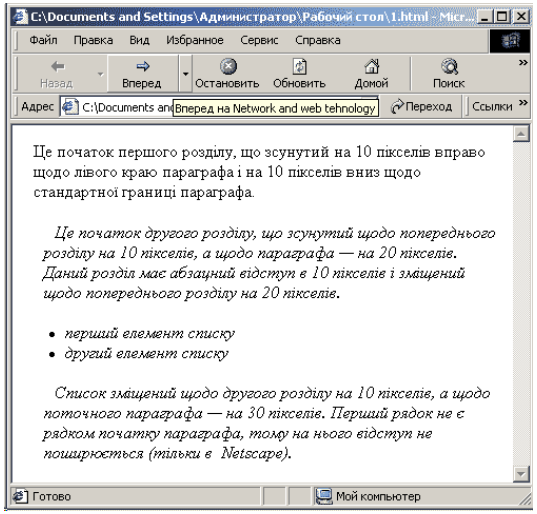


Рис. 1.3 Наслідування і перевизначення

```
<DIV STYLE="margin-left:10px; margin-top:10px;">
```

Це початок першого розділу, що зсунутий на 10 пікселів вправо щодо лівого краю параграфа і на 10 пікселів вниз щодо стандартної границі параграфа.

```
<DIV STYLE="margin-left:10px;margin-top:20px;text-indent:10px;font-style:italic;">
```

Це початок другого розділу, що зсунутий щодо попереднього розділу на 10 пікселів,
а щодо параграфа — на 20 пікселів. Даний розділ має абзацний відступ в 10 пікселів і зміщений щодо попереднього розділу на 20 пікселів.

```
<UL STYLE="margin-left:10px;">
<LI>перший елемент списку
<LI>другий елемент списку
</UL>
```

Список зміщений щодо другого розділу на 10 пікселів,
а щодо поточного параграфа — на 30 пікселів.
Перший рядок не є рядком початку параграфа, тому на нього відступ не поширюється (тільки в Netscape).

```
</DIV>
</DIV>
```

При використанні стилів діють наступні правила старшинства стилів:

Скрипт електронної пошти:

```
$run = $_Get[‘num’]
  If ($_Get[‘read’]){
            …..текст повідомлення
  exit
}
If ($_Get[‘delete’]){
            …..видаляємо
  exit
}
```

Хід виконання роботи

1. Всі наступні завдання виконуйте у каталозі на своєму ftp-акаунті, наприклад .../**lr6/** . Для виконання завдання потрібно мати власну електронну пошту на будь-якій поштовій службі.

2. Створіть скрипт base64.php і дослідіть роботу функції base64_encode. Спочатку надайте значення її аргументу окремих символів, наприклад, “! &*, пробіл, крапка”, потім окремих цифр і на кінець, напишіть фразу. Кодовані рядки спостерігайте у WEB-браузері. Результати занесіть у вигляді таблиці у блог.

```
<html><head><meta content='text/html; charset=UTF-8' http-equiv='content-type'></head><body>
<?php
 // дослідити роботу функції base64_encode
$x = 'Біржовий тренд';
$y = base64_encode($x);
print "$y";
?>
```

3. Зайдіть на сайт своєї поштової служби і знайдіть там інструкції щодо налаштувань поштового клієнта (напр. Outlook Express або Mozilla Thunerbird або The Bat!). З налаштувань з'ясуйте назву SMTP-сервера, його порт і, можливо, якщо такий оговорений, протокол з'єднання, наприклад **ssl** або **tls**.

4. Створіть скрипт для дослідження протоколу надсилання електронної пошти згідно прикладу, що наведений нижче. У відповідних рядках функції **MySendMail** пропишіть назву свого поштового SMTP-сервера, його порт і протокол. В якості поштової скрині **$to** вкажіть свій email, а в якості зворотньої адреси **$from** — будь-який відомий вам інший email. Добийтесь того, щоби ваш скрипт успішно надсилав лист і ви могли його одержати на своїй поштовій службі.

– робимо операції з листами;

– операції з папками (створення, редагування, видалення, редагування);

– закриття з'єднання.

$c = imap-open (рядок з з'єднанням, логін, пароль);
{Рядок з'єднання з сервером: порт/протокол (po3, ssl)}
{192.168.1.9: 110/pop3} inbox
{mail.ua: 113} inbox
{mail.com: 993/IMAP/SSL} inbox
{imap.gmail.com: 993/imap/ssl/novi-date-cerst} inbox

$y = Imap-listmailbox (канал, рядок з'єднання, маска);
$m = imap-headers (…); //в $m потрапить асоціативний масив із списком листів, щоб його роздрукувати використовуємо цикл:

```
For     echo $m as $val
        echo $val."<br>";
```

$g = imap-body (канал, номер листа)

Print "$b"; ця команда видає текст листа у вигляді коду
$bd = imap-qprnt($b);
Print "$bd";
Imap-delete (канал, індекс);
Imap-expange (канал) – видаляє всі листи назавжди;
$s=imapmaillistbox-info (канал) – видаляє статистику на imap-сервері.

Статистка IMAP-серверу:
1. дата;
2. протокол;
3. скриня
4. кількість листів в скрині;
5. останні листи, що надійшли;
6. непрочитані листи;
7. розмір скрині.

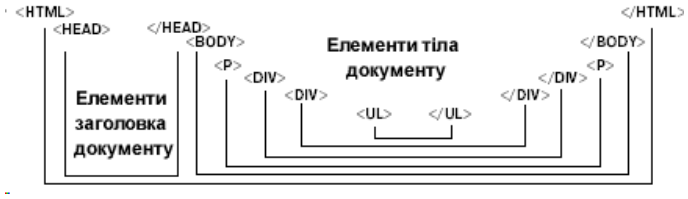


Рис. 1.4 Правила старшинства стилів

- спочатку застосовуються стилі браузера по замовчуванню;
- стилі браузера по замовчуванню перевизначаються стилями вказаними в елементі LINK (елемент заголовка документа);
- прилінкові стилі перевизначаються описами стилів в елементі STYLE;
- стилі елемента STYLE перевизначаються атрибутом STYLE у кожному з елементів розмітки.

Блокові і стрічкові елементи

Пояснити відмінності між блоковим і стрічковими елементами можна на прикладі:

- **параграф** – це блоковий елемент розмітки;
- **виділення курсивом** – це стрічковий елемент розмітки.

Блокові елементи можна вкладати один в одного, але вони не повинні перетинатися. Стрічкові елементи можна як вкладати, так і перетинати, але останнє робити не рекомендується.

Спрощено можна сказати, що атрибути опису стилю стрічкового елемента є підмножиною атрибутів опису стилю блокового елемента.

Узагальненнями блокового і стрічкового елементів , з точки зору стилів, є елементи DIV і SPAN, відповідно.

Елемент DIV

DIV дозволяє застосувати атрибути стилю, пов'язані з границею блоку і відступами блоку від границь старшого елемента, а також "набивка", тобто відступ від границі блоку до границі вкладеного елемента:

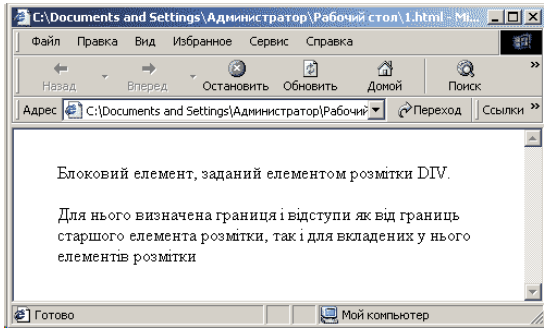


Рис. 1.5 Елемент DIV

```
<DIV STYLE="margin:20px;padding:10px;">
    Блоковий елемент, заданий елементом розмітки DIV.
<P>Для нього визначена границя і відступи як від границь старшого
    елемента розмітки, так і для вкладених у нього елементів розмітки
</P>
</DIV>
```

У даному прикладі всередині вікна браузера розташований блоковий елемент (DIV), всередину якого поміщений ще один блоковий елемент(P). DIV має біле тло і границю.

Елемент SPAN

Елемент розмітки SPAN – це узагальнений стрічковий елемент розмітки, застосування якого не призводить до утворення блоку тексту. Він може замінити елементи FONT, I, B, U, SUB, SUP і т.п. Наведемо приклади таких відповідностей:

Таблиця 1 – Елемент розмітки SPAN

HTML-елемент	CSS-аналог
<FONT COLOR=red> ...</FONT>	<SPAN STYLE="color:red; ">...</SPAN>
<I>...</I>	<SPAN STYLE="font-style:italic; ">...</SPAN>
<B>...</B>	<SPAN STYLE="font-weight:bold; ">...</SPAN>
<U>...</U>	<SPAN STYLE="text-decoration:underline; ">...</SPAN>

Заголовок:
Mime-Version 1.0
Content-type text/html; CHARSET=UTF-8
To: my@i.ua
From: nat@gmail.com
Subject: …
Повідомлення
….
•

MIME – перетворення будь-яких символів у текстовий формат.

```
Function MySendMail ($from, $namefrom, $to, $subject, $message);
$sc = fsockopen ($smtpServer, $port, $errno, $errstr, $timeout);
$sr = fgets($sc, 4096); // читаємо блок з 4096 байтів
fputs($sc, "HELO $localhost". $CR);
$sr = fgets($sc, 4096); // читаємо блок з 4096 байтів
```

fsockopen – відкрити канал зв'язку
fgets – подивитись, що надійшло через socket
fputs – покласти (передати)

```
fputs($sc,"AUTH LOGIN" . $CR);
fputs($sc, base64_encode($username) . $CR);
fputs($sc, base64_encode($password) . $CR);
fputs($sc, "MAIL FROM: <$from>" . $CR);
fputs($sc, "RCPT TO: <$to>" . $CR);
fputs($sc, "DATA" . $CR);
$headers = "MIME-Version: 1.0" . $CR;
$headers .= "Content-type: text/html; charset=utf-8" . $CR;
$headers .= "To: $nameto <$to>" . $CR;
fputs($sc, "To: $to\r\nFrom: $from\r\nSubject:
$subject\r\n$headers\r\n\r\n$message\r\n.\r\n");
fputs($sc,"QUIT" . $CR);
fclose($sc);
```

```
$smtpServer = "tls://smtp.gmail.com";
$port = "465";
$login = "igor@gmail.com";
$password = "%r^trffds"; /
$CR = "\r\n";
```

Порядок роботи з IMAP-сервером:

– під'єднатися до серверу (назва серверу, порт, логін, пароль, протокол);
– одержуємо список поштових скринь;
– одержуємо список листів у вибраній поштовій скрині;

Основні команди з IMAP сервером:

– login;
– password;
– select;
– create;
– rename;
– append(додати листа до поштової скрині);
– search;
– fetch – прочитати лист (завантажити).

Переваги:

– листи зберігаються на поштовому сервері;
– можлива одночасна робота багатьох користувачів з однією скринею;
– підтримка поштових скринь;
– можливість копіювати, видаляти листи без переносу на локальний комп'ютер;
– статус листів і його зміна відбувається на сервері;
– можливість пошуку інформації засобами серверу;
– онлайн режим.

Задача. Створити скрипт для надсилання електронної пошти.

Кому	
Від кого	
Текст	

Діалог поштового скрипту з SMTP-севером:

– з'днання з SMTP-севером через порт 25, 465;
– надсилання команди Hello;
– надсилання команди AUTH LOGIN (авторизація)
– надсилаємо логін у формі MIME
– надсилаємо пароль у формі MIME
– надсилаємо команду MAIL FROM і свою ел. адресу;
– надсилаємо текст листа, після команди Data;
– в кінці листа надсилаємо крапку •;
– надсилаємо QUIT (прощаємось з роботою).

Властивості блоків

Блокові елементи (блоки тексту або box) дозволяють оперувати з текстом у термінах прямокутників, що займає цей текст. При цьому блок тексту стає елементом дизайну сторінки з тими ж властивостями, що і картинка, таблиця або прямокутна область додатка.

Блок тексту має такі властивості: висоту (height), ширину (width), границі (border), відступ (margin), набивка (padding), довільне розміщення (float), керування обтіканням (clear).

Графічно властивості можна представити наступним чином:

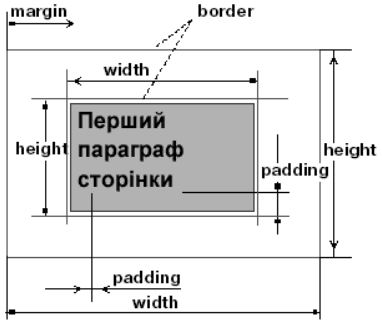


Рис. 1.6 Властивості блоків

Ширина і висота блоку тексту може задаватися у типографських пунктах (pt), пікселях (px) і умовних одиницях (em)

Відстань від границі блокового елемента до границі вкладеного в нього блокового елемента називається padding або "набивка", або словосполучення "внутрішній відступ".

Відступ від "набивки" зовнішнього блокового елемента до границі вкладеного елемента називається margin. Для його позначення ми будемо вживати термін "відступ" або словосполучення "зовнішній відступ".

У такий спосіб padding і margin характеризують відступи блокового елемента (див рис. 1.6).

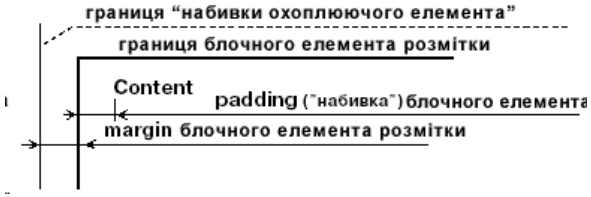


Рис. 1.7

Відступи і "набивки" можуть бути лівими, правими, верхніми і нижніми.

Відступи (margin)

При відображенні блоку тексту на папері довкола нього зазвичай залишають поля. Поля можна задавати або щодо границі сторінки, або щодо самого блоку тексту. У першому випадку ми маємо справу з "набивкою" (padding), а в другому — з відступом (margin). Власне, ширина поля буде визначатися сумою ширини "набивки" і ширини відступу.

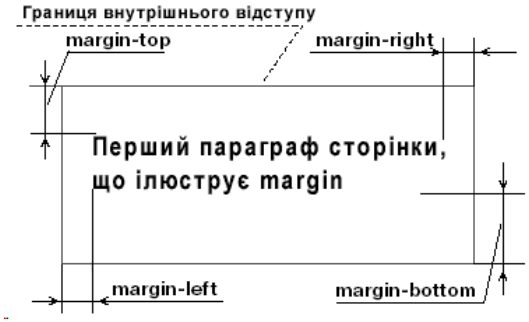


Рис. 1.8 Відступи (margin)

Пунктирна лінія і границя блоку є невидимими лініями, тобто уявними по вирівняному краю тексту (сумарна ширина полів). Стрілки вказують напрямок відліку відступу. Padding відраховується від зовнішньої границі блоку всередину блоку, у той час як margin – від зовнішньої границі блоку в область охопленого ним блоку (назовні).

Зовнішній відступ (margin) може відраховуватися по будь-якому напрямку щодо сторін блоку:

ЛАБОРАТОРНА РОБОТА № 6

Тема: Дослідження взаємодії WEB-проектів із SMTP-серверами при надсиланні електронної пошти

Мета: Дослідження методів надсилання електронної пошти через SMTP-сервери у WEB-проектах

Теоретичні відомості

SMTP(Simple mail transfer protocol) – сервер надсилання пошти (port 25). Sendmail – програма для надсилання пошти.

Функції:

1. одержання команд аутентифікації від поштової програми;
2. одержання листів;
3. закриття сеансу.

POP (Past office protocol) – сервер одержання пошти. (port 110)

Функції:

1. аутентифікація;
2. одержання листів поштовою програмою;
3. знищення листів на поштовому сервері;
4. роз’єднання.

Протокол POP3 забезпечує передачу листів на комп’ютер й вся подальша робота відбувається в поштовій програмі (пошук, читання).

Недоліком є те,що без завантаженого листа неможливо прочитати його заголовок.

IMAP (Internet message Access Protocol) – протокол доступу до повідомлення. Працює аналогічно до POP3, але дозволяє:

1. здійснити пошук за ключовими словами засобами поштового серверу;
2. дозволяє не забирати пошту на локальний комп’ютер
3. управляння листами (одержання, надсилання), поштовими скринями можна здійснювати без пересилання листів.
4. при приєднання до серверу користувач бачить лише заголовки листів.

Контрольні запитання:

1. Що таке хеш-функція?
2. Що означає незворотність перетворення хеш-функції?
3. Що означає стійкість до колізій 1-го роду хеш-функції?
4. Що означає стійкість до колізій 2-го роду хеш-функції?
5. Що означає “лавинний ефект” у хеш-функції?
6. Із скількох байтів складається хеш, що створюється за методом md5?
7. Чому паролі повинні складатись із багатьох символів?
8. Яка найменша рекомендована довжина паролю і чому?
9. В чому полягає метод **brute force**?
10. Чому можна швидко розкрити паролі із стандартних слів навіть якщо вони мають велику довжину?
11. З яких символів (або їх комбінації) повинні складатись паролі, щоби до них було важко застосувати метод **brute force** або **rainbow**?
12. В чому полягає метод формування хешів **salt**?

- **margin-left** – лівий зовнішній відступ. Визначає відстань від лівої границі блоку тексту до лівої границі внутрішнього відступу ("набивки", padding) охопленого елемента;
- **margin-right** – правий зовнішній відступ;
- **margin-top** – верхній зовнішній відступ;
- **margin-bottom** – нижній зовнішній відступ;
- **margin** – задає загальний зовнішній відступ від усіх сторін блоку тексту. Застосовується в тому випадку, якщо блок тексту равновіддалений від усіх границь внутрішнього відступу охопленого елемента.

Рис. 1.9

Якщо розмір усіх зовнішніх відступів однаковий, то можна просто скористатися атрибутом margin:

```
P { margin:5px; }
```

Набивка (padding)

Між границею і змістом блоку є вільний простір. Він називається внутрішній відступ текстового блоку або padding. Разом із зовнішнім відступом (margin) текстового блоку padding утворить загальне поле відступу від границі охопленого блокового елемента розмітки.

У блоці тексту існує чотири сторони. Відповідно, padding може бути:

- **padding-left** – лівий внутрішній відступ, що визначає відстань від лівого краю блоку до його змісту;
- **padding-right** – правий внутрішній відступ;

- **padding-top** – верхній внутрішній відступ;
- **padding-bottom** – нижній внутрішній відступ;
- **padding** – визначає єдиний розмір внутрішнього відступу блоку. Цей параметр задається у випадку однакового розміру відступу від усіх сторін блоку.

```
P { padding-left:100px; padding-right:50px;
    padding-top:20px; padding-bottom:10px;
    text-align:left; border-width:1px; }
```

Рис. 1.10

Границя (border)

У кожного блокового елемента розмітки є границя. Від границі відраховуються відступи (margin і padding). Вздовж границі "плаваючого" блоку його обтікає текст.

Для опису границь блоків застосовуються наступні атрибути:

- **border-top-width** – ширина верхньої границі блоку;
- **border-bottom-width** – ширина нижньої границі блоку;
- **border-left-width** – ширина лівої границі блоку;
- **border-right-width** – ширина правої границі блоку;
- **border-width** – ширина границі блоку. Задається в тому випадку, якщо ширина границі блоку однакова по всьому периметрі блоку;
- **border-color** – колір границі блоку. Відповідно до специфікації CSS1 може бути заданий для кожної з границь блоку. Наприклад, border-right-color:red. Може задаватися як мнемонікою (red, blue, navy і т.п.), так і в нотації RGB (border-color:#003366);
- **border-style** – тип лінії границі блоку. Може приймати значення: none, dotted, dashed, solid, double, groove, ridge, inset,

і одержані хеші з аргументами занотуйте у порівняльну таблицю звіт (структуру таблиці див. нижче).

```
<?php
        $x = '111'; // аргумент функції
        $h = md5($x); // хеш
        print "$h<br>";
?>
```

3. *Завдання 2.* Додайте до скрипту код, щоби хеші обраховувались від сумарного рядку, який складається з аргументу і символів «hash». Обрахуйте хеші до аргументів, наведених у п.2. Дані занесіть у порівняльну таблицю у звіт.

4. *Завдання 3.* Додайте до скрипту код, щоби хеші обраховувались від сумарного рядку, який складається з аргументу і перших двох символів агрументу (метод **salt**). Обрахуйте хеші до аргументів, наведених у п.2. Дані занесіть у порівняльну таблицю у звіт.

5. *Завдання 4.* Скопіюйте у каталог .../lr5/task1/ файли з лабораторної роботи N4 і доопрацюйте їх таки чином, щоби в аутентифікація користувачів відбувалась за хешами паролів. Спосіб створення хешів оберіть **salt**.

6. *Завдання 5.* Результати досліджень опишіть у своєму блозі. Зверніть увагу на ступінь відмінності хешів із близькими аргументами.

Таблиця 1 – Порівняння хешів

N	Пароль	md5($x)	md5($x.'hash')	md5($x.$salt)
1	1			
2	2			
3	ABCDefg			
4	ABCDefh			

ЛАБОРАТОРНА РОБОТА № 5

Тема: Засоби безпеки у WEB-програмах. Хеш-функції.

Мета: Дослідження методів шифрування і використання їх у WEB-проектах.

Теоретичні відомості

Хеш функція — функція, що перетворює вхідні дані будь-якого (як правило, великого) розміру в дані фіксованого розміру.

Криптографічна хеш-функція повинна забезпечувати:

- стійкість до колізій (два різні набори даних повинні мати різні результати перетворення);
- необоротність (неможливість обчислити вхідні дані за результатом перетворення).

Хеш-функції також використовуються в деяких структурах даних – хеш таблицях і декартових деревах. Вимоги до хеш-функції в цьому разі інші:

- добра перемішуваність даних;
- швидкий алгоритм обчислення.

Список алгоритмів

- HAVAL
- MD2
- MD4
- MD5
- N-Hash
- RIPEMD-160
- SHA
- Snefru
- Tiger
- Whirlpool

Хід виконання роботи

*1. Всі наступні завдання виконуйте у окремих каталогах на своєму ftp-акаунті, наприклад .../**lr5/task1/***

2. *Завдання 1.* Створіть скрипт для дослідження функції для створення hash — md5. В якості аргументів послідовно надайте « », «1», «2», «ABCDefg» та «ABCDefh». Порівняйте одержані хеші. Висновки

outset. Відповідно до специфікації CSS1, може бути заданий для кожної з границь блоку. Наприклад, border-right-style:dotted. Вказівка типу лінії границі підтримується не всіма браузерами.

Наприклад, для опису верхньої лінії границі можна використовувати запис типу:

```
P { border-top:1px dotted red; }
  атрибут: ширина_лінії тип_лінії колір_лінії код
```

Якщо необхідно обмежити блок тексту границею, то це може виглядати приблизно так:

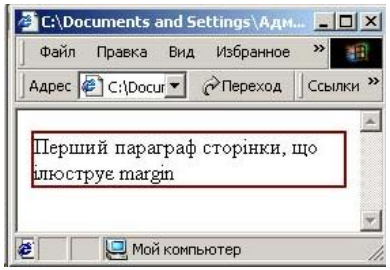


Рис. 1. 11

У цьому прикладі ми використовували наступний опис стилю відображення границі:

```
P {text-align:left;border-width:2px;border-color:darkred;border-style:solid; }
```

Обтікання блоку тексту

Під обтіканням блоку текстом розуміють той самий ефект, який можна досягнути для графіки, коли картинка не розриває блок тексту, а вбудовується в нього.

Обтіканням блоку тексту іншим текстом керують два атрибути CSS: float і clear.

Атрибут float визначає "плаваючий" блок тексту. Він може приймати значення:

left – блок притиснутий до лівої границі охоплюючого елемента;

right – блок притиснутий до правої границі охоплюючого елемента;

both – текст може обтікати блок по обидва боки.

Проілюструвати обтікання дозволяє наступний приклад:

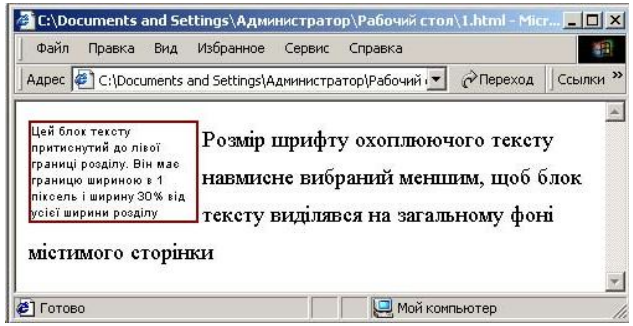


Рис. 1.12 Атрибут опису стилів float

Другий атрибут опису стилів clear не допускає наявності "плаваючих" блоків біля блоку тексту. Атрибут може приймати начення: right, left, none, both:

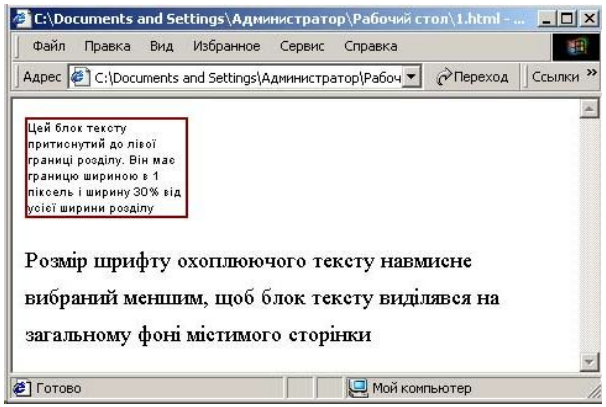


Рис. 1.13 Атрибут опису стилів clear

Хід виконання роботи:

1. Завантажити операційну систему Linux.
2. Запустити файловий менеджер (ФМ) Krusader і приєднатись по **ftp** до сервера, використовуючи логін і пароль свого власного **ftp**-облікового запису. В з'єднанні використати сервер **192.168.1.5** або **vsau.vin.ua** (Нагадаємо, що для вмикання **ftp**-з'єднання в файловому менеджері Krusader треба або натиснути Ctrl_N або ж скористатись його меню).

5. Встановіть термін дії куки вашого скрипту 20 сек. і дослідіть, як це вливає на роботу. Що відбувається із кукою в браузері після спливу терміну її дії? Занотуйте у звіт свої спостереження.
6. *Завдання 2.* Перепишіть створені у попередній лабораторній роботі скрипти для аутентифікації користувача у каталог ***/lr4/task2/*** і доопрацюйте їх таким чином, щоби скрипт запам'ятовував останнього користувача і при завантаженні сторінки аутентифікації відбувався автоматичний перехід на робочу сторінку з виведенням вітання.
7. *Завдання 3.* Перепишіть у каталог ***/lr4/task3/*** скрипт із електронним java-календарем з курсу ТСПІС і доопрацюйте його таким чином, щоби календар міг запам'ятовувати останню встановлену на ньому дату.
8. Результати досліджень опишіть у своєму блозі.

Контрольні запитання:

1. Що таке COOKIE і для чого вони використовуються?
2. Які поля мають COOKIE і для чого вони призначені?
3. Що спільного і різного між COOKIE і сесіями?
4. Чи можна користуватись новим значенням COOKIE у скрипті, що його встановлює і чому? Перевірте на простому скрипті.
5. Що відбувається з файлом COOKIE після того як спливе час його дії?
6. Чи можна з одного і того ж сервера надіслати декілька COOKIE з різними значеннями індексного поля? Чи це буде один COOKIE-файл чи декілька?
7. Чи можна здійснити передачу інформації між WEB-сторінками за допомогою COOKIE? Якщо так, то поясніть на власному прикладі.
8. Яку перевагу мають COOKIE перед SESSIONS, GET та POST?
9. Які недоліки мають COOKIE?
10. За допомогою якого оператора можна перевірити, чи ввімкнені COOKIE у WEB-браузері?

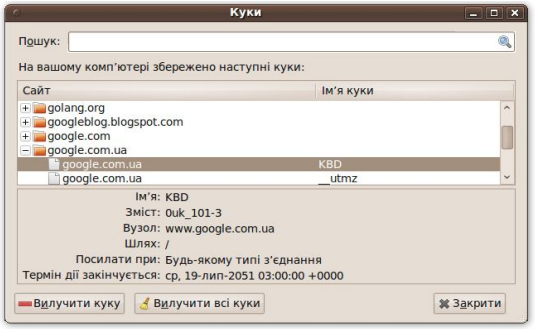


Рис. 4.1

Відкрийте теку будь-якого сайту і дослідіть значення полів одержаних кук. Вилучіть всі куки і після цього зайдіть на будь-які власноруч обрані сайті. Перевірте, чи були браузером одержані куки і який у них зміст. Дослідження знотуйте у звіт.

3. Створити php-скрипт для дослідження відвідування сторінки. Текст скрипта взяти із лекційного матеріалу. Термін дії встановіть 10000 сек. Послідовно перевантажуючи сторінку, переконатись, що лічильник відвідувань працює. Користуючись методами п.2, дослідіть куки вашого скрипту. Перегляньте зміст своїх кук після кожного перевантаження сторінки.

4. Створіть скрипт для перевірки наявності увімкнених COOKIE:

```
<?php
        //перевірка, чи ввімкнені cookie у WEB-браузері
        if(!$_GET['x'])
        {
                // перевантажуємо цей скрипт з параметром x=1 для
  перевірки, чи встановлені cookie
                header("Location: $PHP_SELF?x=1");
                // встановлюємо cookie с полем "check"
                setcookie("check","1");
        }
        else
        {
         if(!$_COOKIE['check']) {
                print "Cookie в браузері не ввімкнені !";
        } else {
                print "Cookie ввімкнені. Можна працювати.";
                //header("Location: abc.php");
                // попередній оператор можна розкоментувати. Тоді
  буде здійснюватись перехід на необхідну сторінку.
                }
        }
?>
```


3. Після успішного з'єднання із сервером, користуючись меню файлового менеджера, створіть каталог **7-sem**, у ньому підкаталог **lr1,** а у ньому новий файл з назвою **lr1.html**. Одразу після цього має відкритись текстовий редактор. Збережіть цей порожній текст (Ctrl_S) і далі внесіть до нього наступний HTML-код. Відкрийте цей файл WEB-браузером:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=utf-8">
</head>

<body>
<html>
   <h1>Дослідження CSS. Викона(ла)в студ. гр. 41-ЕК ...</h1>
</html>
</body>
```

4. На робочому столі створіть каталог (теку) **css_img** для файлів із зображеннями екрану. Натисніть на клавіатурі кнопку **PrtScr** і збережіть зображення екрану з браузером у теці **css_img** під назвою **1.png** .

5. Відкрийте це зображення програмою GIMP, виділіть область із заголовком “Дослідження CSS... ”. Скопіюйте цю область в буфер. Користуючись меню “Файл” -> “Сворити” -> “З буферу обміну (Shift-Ctrl-V)” створити зображення заголовку і зберегти у файл **1-1.png**.

6. Відкрийте у новій вкладці браузера свій персональний кабінет і увійдіть до блогової системи університету, користуючись своїм обліковим записом. Оберіть пункт меню “Мої сайти” та створіть сайт з назвою **wxxxxx**, де **xxxxx** — ваш логін. Назву сайту надайте “WEB-дизайн (ПІБ)”.

Після успішного створення власного сайту перейдіть до пункту “Керування” і далі до пункту “Мої сайти”. Виберіть рядок із назвою щойно створеного сайту і перейдіть до пункту “Майстерня”.

Перейдіть до пункту “Публікації” і натисніть “Додати”. Вам відкриється вікно редактора, у якому має створюватись блог. Введіть заголовок “Дослідження CSS студента (ПІБ).” Опишіть своїми словами тему, ціль і мету дослідження. Напишіть короткі теоретичні відомості. Наведіть HTML-текст

сторінки з п.3 і додайте до нього нього зображення з файлу **1-1.png**. Для цього над вікном редактора скористайтесь гіперпосиланням “Завантажити\Вставити”. Після завантаження у полі назва введіть текст, що описує зображення на цьому файлі, наприклад, “Тег h1 без CSS”. Після завершення редагування натисніть кнопку “Чернетка” (Увага! Не натискайте кнопку “Опублікувати” до повної готовності блогу!).

У подальших пунктах лабораторної роботи продовжуйте послідовно досліджувати роботу CSS і кожен з них аналогічно описуйте у своєму блозі.

При входженні до своїх блогів не забувайте спочатку заходити до пункту меню “Мої сайти”, вибирати свій сайт і саме там заходити до пункту “Майстерня”. Якщо ви це робитиме, то ваш блог буде опублікований на головній сторінці блогосфери ВНАУ, а не у приватному блозі.

7. Додайте до коду WEB-сторінки наступні рядки, збережіть файл і оновіть зображення сторінки у WEB-браузері. Збережіть зображення екрану з браузером у теці css_img під назвою **2.png.** Наведіть у своєму блозі текст введеного стилю і одержане зображення з власними коментарями:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
charset=utf-8">
</head>
<body>
<style>
h1 {
    color: red;
    }
</style>
<html>
<h1>Дослідження CSS. Викона(ла)в студ. гр. 31-ЕК ...</h1>
</html>
</body>
```

8. *Дослідження властивостей шрифтів і фону.* Послідовно додайте до стилю h1 властивості font-size, font-family і т.д. і кожну з них разом з відповідним рисунком занотуйте у своєму блозі (дослідіть також і закоментовані опції):

1. Взнати скільки секунд минуло з 1 січня 1970 року.
2. Визначити у секундах тривалість періоду часу, у перебігу якого файл “cookie” буде діяти.
3. Скласти кількість секунд, які минули з 1 січня 1970 року з кількістю секунд, які складають термін дії файлу “cookie”.

Пам’ятайте, що у добі 86400 секунд (60x60x24). Ось код необхідної стрічки:

```
setcookie ("username ", "mukola", time() +(86400 * 30));
```

Функція setcookie () поміщає файл “cookie” у комп’ютер користувача на термін 30 днів. В любий час на протязі 30 днів можна отримати доступ до змінної $username зі сценарію і змінна поверне (в наведеному вище прикладі) значення "mukola".

Для видалення файлу “cookie” достатньо створити інший подібний файл з такою ж назвою, але не надавати значення змінній і час збереження файлу. В наступному прикладі наведено, як видалити вже створений файл “cookie”.

```
setcookie ("username ");
```

Важливо відмітити, що файли “cookie” потрібно встановлювати перед тим як відправити будь-який текст у браузер (тобто виклик функції setcookie() повинен передувати тегам <HTML> або <HEAD>). При спробі створити файл “cookie” після передачі тексту у браузер виводиться повідомлення про помилку, яке попереджує, що потрібний файл не був створений.

Хід виконання роботи

1. *Всі наступні завдання виконуйте у окремих каталогах на своєму ftp-акаунті, наприклад .../**lr4/task1/***

2. *Завдання 1.* Дослідити куки у WEB-браузері. Увімкнути WEB-браузер (надалі приклад розглядається на прикладі Mozilla Firefox) і завантажити порожню сторінку. Зайти у пункт меню «Інструменти»-«Інформація про сторінку» і обрати вибрати підпункт «Безпека». Вибрати пункт «Переглянути куки». Ви маєте побачити таблицю із списком сайтів, які раніше відкривались у браузері і список кук, що зберігаються на вашому комп'ютері (Рис.4.1).

ЛАБОРАТОРНА РОБОТА № 4

Тема: Дослідження COOKIE.

Мета: Дослідження особливостей використання COOKIE і набуття навичок у застосуванні їх у WEB-проектах.

Теоретичні відомості

Функція setcookie ()

setcookie (name [, value [, expiration]]);

Ця функція дозволяє зберегти деяку інформацію, яку називають "cookie", в комп'ютері користувача. Файл "cookie" – це невеликий файл, який містить любі необхідні вам відомості. Зазвичай "cookie" використовують для зберігання інформації про користувача або його налаштуваннях.

Функція setcookie () застосовується у двох варіантах:

- створення файлів "cookie", які діють на протязі сеансу;
- створення файлів "cookie", які мають певний термін дії.

Файли "cookie", які діють на протязі сеансу

Такі файли зберігаються на комп'ютері користувача до тих пір, поки браузер не закінчить свою роботу. Після того, як користувач закриває браузер, файл "cookie" знищується.

Для створення файлу "cookie", який діє на протязі сеансу, достатньо викликати в сценарії функцію setcookie () та передати їй ім'я змінної та значення.

```
setcookie ("username ", "mukola");
//еквівалентно виразу $username = mukola
```

Після створення файлу "cookie", змінну, яка в ньому міститься можна використовувати на любій сторінці вузла. Вона буде доступна до тих пір, поки користувач не закриє браузер.

Створення файлів "cookie", які мають певний термін дії

Файл, термін дії якого закінчується через визначений проміжок часу, створюється аналогічно файлу "cookie", який існує на протязі сеансу. Єдине виключення – необхідно указати дату закінчення строку його дії. Ця дата задається у секундах, починаючи з 1 січня 1970 року. Ось для чого потрібна функція time(), яка повертає кількість секунд, які пройшли з 1 січня 1970 року.

Якщо ви хочете створити файл "cookie", термін дії якого закінчиться за 30 днів, вам потрібно виконати наступне:


```
h1 {
    color: red;
    background: #f0f0f0;
    font-style: italic; /*oblique*/
    font-weight: normal; /*bold normal*/
    font-variant: small-caps; /*normal*/
    font-size: 10pt;
    font-height: 10.2em;
    font-family: monospace; /*sans-serif serif fantasy*/
    font-size: 10pt;
    }
```

9. *Дослідження властивостей відступів і інтервалів.* Послідовно додайте до стилю **p** властивості **background**, **padding** і т.д. і кожну з них разом з відповідним рисунком занотуйте у своєму блозі (дослідіть також і закоментовані опції):

```
p {
  background: silver;
  padding: 0pt; /* 10 ex */
  margin-left: 100pt;
  margin-right: 500pt;
  margin-top: 20pt;
  margin-bottom: 24pt;
  /* margin: 2px 500pt 1pt 24pt; */
  font-size: 10pt;
  }
.....
<body>
<p>Сто років мучених надій,</p>
<p>і сподівань, і вір, і крові</p>
<p>синів, що за любов тавровані,</p>
<p>сто серць, як сто палахкотінь.</p>
.....
```

10. *Дослідження властивостей рамок і розташування тексту.* Після останнього тегу </p> додайте два вкладених шари з текстом. (Горизонтальні лінії <hr> додані в код спеціально для того, щоби можна було дослідити роботу **margin**):

```
<hr>
<div id='d1'>
<hr>
<div id='d2'>
Слідкуйте за новим цікавим вмістом в Інтернеті.
</div>
</div>
```

іЗверніть увагу на те, що шари <div...> мають ідентифікатори **d1** і **d2**. Їх і будемо використовувати в якості селекторів стилів. Спочатку додайте стиль до шару **d1.** Зверніть увагу на те, що поки до шару **d2** ніяких стилів не застосовується**:**

```
#d1 {
    margin: 20%; /* Відступи навколо цього елементу */
    background: #fd0; /* Колір фону */
    padding: 10px; /* Поля навколо тексту */
    }
```

Перегляньте результат у браузері і занотуйте у блозі його опис. Послідовно змініть значення **margin** і **padding.** Користуючись положенням горизонтальних ліній, з'ясуйте, на що саме впливають ці опції.

11. Додайте стиль для шару d2, перегляньте одержаний результат у браузері, проаналізуйте їх призначення і вплив на роботу. Підставте і дослідіть вплив різних значень опцій шляхом зміни величин біля них:

```
#d2 {
    border-width: 20px; /* Ширина рамки */
    border-color: #666; /* Колір рамки */
    border-style: ridge; /* Параметри рамки solid dashed dotted double
groove ridge inset outset*/
    padding: 10px; /* Поля навколо текста */
    margin: 20px; /* Відступи навколо */
    text-align: left; /* Розташування тексту center left right justify */
    text-indent: 2em; /* Відступ першого рядку в абзаці */
    text-decoration: underline; /*Управл. властивостями тексту blink line-
through overline underline none */
    }
```

Після завершення експериментів з ***padding*** *і* ***margin*** *приберіть теги <hr>.*

12. *Дослідження властивостей шрифтів.* Додайте до вашої WEB-сторінки шар з ідентифікаторм d3 і тектстом “Дослідження ШРИФТІВ”, а також стиль для до цього шару. Як і раніше, поступово додавайте до стилю по одному рядку

у цій секції має встановлюватись сесія '**user**' і виводитись гіперпосилання на сторінку **bank.php** з текстом: “Користувач зареєстрований. Розпочати роботу.”. Щоби після повідомлення не виводилась знову форма, використайте оператор завершення виконання коду **exit;**

Контрольні запитання:

1. У чому різниця методів GET і POST?
2. Чи можна не використовувати форму для методу GET? Якщо так, то наведіть приклади.
3. Чи можна не використовувати форму для методу POST? Якщо так, то наведіть приклади.
4. Чи можна не використовувати форму для методу SESSION? Якщо так, то наведіть приклади.
5. Поясніть, що передає кожен елемент WEB-форми **hidden**, **text**, **radio**, **password**, **textarea**, **checkbox**, **select**, **reset** і **submit** своєму “нащадку”
6. Що таке сесія і як вона може використовуватись для передачі даних між сторінками?
7. Чи є сесія спільною для різних WEB-браузерів одного типа (наприклад, Firefox), якщо вони працюють на одній IP-адресі або комп'ютері? Поясніть чому?
8. Яка обов'язкова вимога до розташування оператора **session_start();** у скрипті і чому?
9. Поясніть як сесіям надаються значення і як їх можна зчитувати?
10. Поясніть чому web-форма, яка надає значення сесіям (наприклад, **login.php)** повинна мати рекурсію або іншу допоміжну сторінку?
11. Якщо користувач працює із сервером і при цьому використовує сесії, в яких передаються його конфіденційні дані, наприклад пароль, то чи можна з іншого браузера чи програми їх зчитати?
12. Порівняйте переваги і недоліки методів GET, POST і SESSIONS. Наведіть приклади, де який з них має переваги перед іншими.

15. У першому WEB-браузері відкрийте у третій закладці скрипт **sess_stop_name.php**. Після цього перейдіть на закладку цього браузера із **sess_get.php.** Оновіть цю сторінку. Що ви спостерігаєте?

16. Перевірте, що змінилось у сесіях у другому WEB-браузері? Для цього там також оновіть сторінку із **sess_get.php.** Занотуйте висновки у звіт.

17. Перейдіть у перший WEB-браузер і у наступній закладці активізуйте скрипт **sess_stop_all.php**. Аналогічно у цьому і другому браузерах перевірте, що відбулось із сесіями. Занотуйте спостереження у свій звіт.

18. Дослідіть роботу оператора **session_start();** Для цього встановіть знову сесії, оновивши сторінку із скриптом **sess_set.php.** Закоментуйте у скрипті **sess_stop_all.php** рядок **session_start();.** Аналогічно проведіть дослідження роботи цього скрипту. Занотуйте висновки у звіт.

19. *Завдання 4.* Створіть прототип скрипту реєстрації користувача **login.php**, який поки буде лише виводити повідомлення “Реєстрація користувача”.

20. Створіть скрипт **bank.php**, який буде аналізувати наявність сесії і, у випадку її відсутності, буде переадресовувати користувача на іншу — **login.php** для проходження реєстрації:

```
<?php
    session_start();
    if(!isset($_SESSION['user'])) {
    header('Location: login.php');
    }
    print “Користувач зареєстрований. Все у порядку. Продовжуємо далі...”;
?>
```

21. Перевірте роботу створених скриптів, коли сесія '**user**' існує і коли ні. (Вмикайте-вимикайте сесію будь-яким із досліджених вище способів).

22. *Завдання 5.* Взявши за основу скрипти **bank.php** і **login.php**, доопрацюйте їх таким чином, щоби у скрипті **login.php** була форма для введення імені користувача. Причому після натискання кнопки типу **submit** цей скрипт повторно запускав сам себе. Перед формою розташуйте секцію, яка б аналізувала, чи була натиснута кнопка типу **submit**. Якщо так, то

з приклада нижче, досліджуючи властивості шрифтів і спостерігаючи зміни тектсту у WEB-браузері:

```
#d3 {
    font-style: normal; /* normal italic oblique */
    font-variant: small-caps;
    font-family: cursive; /* serif sans-serif monospace fantasy cursive */
    font-size: 10px; /* 140% em px pt */
    font-weight: normal; /* normal bold */
    }
```

Опишіть разом із скріншотами у своєму блозі досліджені властивостей шрифтів.

Після повного написання блогу опублікуйте його, натиснувши кнопку “Опублікувати”. Він стане доступним за адресою http://vsau.vin.ua/wp/wxxxxx

Примітка: В позначених синім кольором операторах послідовно змінити значення опцій і для кожної з них занести малюнки з коментарями до них до свого блогу.

Контрольні запитання:

1. Що таке CSS і які вони мають головні властивості?
2. Де можуть зберігатись інструкції CSS?
3. Що таке селектори? Наведіть приклади.
4. Що таке селектори класа? Наведіть приклади.
5. Що таке ID-селектори? Наведіть приклади.
6. Чи може один стиль бути застосованим одночасно до декількох селекторів?
7. Наведіть приклади і поясніть, як працюють селектори контексту.
8. Що таке селектори нащадків? Наведіть приклади.
9. Наведіть базові властивості тексту і приклади використання стилів для їх управління.
10. Наведіть базові властивості шрифтів і приклади використання стилів для їх управління.
11. Наведіть базові властивості рамок і приклади використання стилів для їх управління.
12. Для чого у WEB-стрінках використовується тег ?

ЛАБОРАТОРНА РОБОТА № 2

Тема: Дослідження каскадних стилів CSS. Частина 2.

Мета: Набуття навичок створенні і застосуванні каскадних стилів CSS. Базові властивості

Теоретичні відомості

Керування кольором у CSS

Каскадні таблиці стилів (CSS) у першу чергу описують властивості тексту. Це стосується як текстових блоків, так і стрічкових елементів розмітки змісту сторінки.

Колір тексту

У HTML для керування кольором тексту використовується елемент FONT. Його аналогом у CSS є атрибут color. Цей атрибут можна застосовувати як для блокових, так і для стрічкових елементів розмітки.

Розглянемо як блоковий елемент розмітки комірку таблиці:

```
TD {color:darkred;}
```

При визначенні кольору тексту для блокового елемента весь текст цього елемента відображається заданим кольором. Часткова зміна кольору можлива, якщо помістити стрічковий елемент розмітки всередину блокового:

```
P {color:darkred;}
I  {color:#003366;font-style:normal;}
```

Колір тла тексту

У HTML кольором тла можна керувати тільки для конкретного блокового елемента розмітки. Таким елементом може бути вся сторінка:

```
<BODY BGCOLOR=...>...</BODY>
```

Або, наприклад, таблиця:

```
<TABLE BGCOLOR=...>...</TABLE>
```

У наведеному нижче прикладі (рис. 2.1) для виділення тексту застосоване інвертування кольору тла і кольору тексту:

b). Створіть скрипт **sess_get.php** і внесіть до нього такий текст:

```
<?php
      session_start();

      $x = $_SESSION['id'];
      $y = $_SESSION['user'];
      $z = $_SESSION['e-mail'];

      print "
      id = '$x'<br>
      user = '$y'<br>
      e-mail = '$z'<br>
      ";
?>
```

c). Створіть скрипт **sess_stop_name.php** і внесіть до нього такий текст:

```
<?php
      session_start();
      if(isset($_SESSION['user'])) {
        unset($_SESSION['user']);
        print "Зупинямо сесію user ...";
      } else {
        print "Сесія 'user' порожня.";
      }
?>
```

d). Створіть скрипт **sess_stop_all.php** і внесіть до нього такий текст:

```
<?php
      session_start();
      session_destroy();
      print "Відаляємо всі сесії...<br>";
?>
```

11. *Завдання 3. Увага! Далі все виконувати строго за інструкцією!* Спочатку відкрийте у WEB- браузері скрипт **sess_get.php** і занотуйте результати передачі даних сесіями.

12. У другій вкладці WEB-браузера відкрийте скрипт **sess_set.php.** Перейдіть до вкладки з **sess_get.php** і оновіть сторінку. Що ви спостерігаєте?

13. Відкрийте ще один WEB-браузер і в ньому скрипт **sess_get.php.** Що ви спостерігаєте?

14. Тепер у другій його вкладці також відкрийте скрипт **sess_set.php.** Оновіть сторінку з **sess_get.php**. Що ви спостерігаєте?

3. Встановити на формі метод **GET**. В якості **action** вказати скрипт **get_show.php.**

4. В цьому ж каталозі створити скрипт **get_show.php**. Прописати на його початку оператори для одержання даних від форми скрипту **get_send.php** методом **GET** (типу **$x = $_GET['abc'];**) За допомогою операторів **print** або **echo** виведіть значення змінних на WEB-сторінку.

5. Відкрийте у WEB-браузері скрипт **get_send.php**, заповніть поля форми та інші елементи і натисніть кнопку типу **reset**. Занотуйте у звіт, що відбувається.

6. Повторно заповніть поля форми і натисніть кнопку типу **submit**. Що ви спостерігаєте у командному рядку WEB-браузера? Розберіться, як передались дані до скрипту **get_show.php**? Особливо зверніть увагу на значення елементів типу **radio password, checkbox, select, reset і submit**. Висновки занотуйте у звіт.

7. *Завдання 2.* Скопіюйте скрипт **get_send.php** під назвою **post_send.php**. Встановіть у ньому метод передачі даних **POST**, а в **action** вкажіть скрипт **post_get.php**.

8. Скопіюйте скрипт **get_send.php** під назвою **post_get.php**. Замініть асоціативний масив **$_GET** на **$_POST**.

9. Проведіть дослідження із скриптами **post_send_php** і **post_get.php** згідно п.4. і п.5

10. а). Створіть скрипт **sess_set.php** і внесіть до нього такий текст:

```
<?php
    session_start();

    $_SESSION['id'] = '866';
    $_SESSION['user'] = 'Василенко';
    $_SESSION['e-mail'] = 'vvv@mail.ua';

    print "Надаємо сесіям значення ...";

?>
```



```
<SPAN STYLE="background-color:black;color:white;">
        як стрічкові елементи розмітки
</SPAN>
```

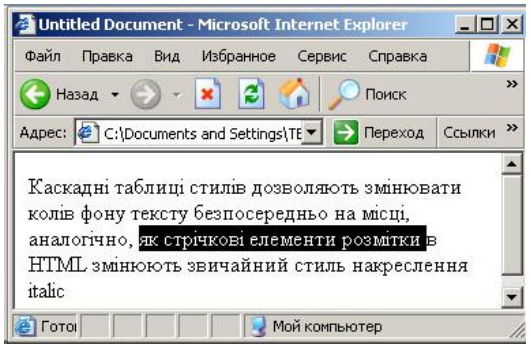


Рис. 2.1 Колір тла тексту

Для роботи з тлом елементів існують атрибути: background-image; background-repeat; background-attachment; background-position. Усі властивості тла можна описати в атрибуті background:

```
background:transparent|color url repeat scroll position
```

Приклад:

```
P{background: gray http://vsau.vin.ua/vsau.gif no-repeat fixed center
  center;}
```

Керування шрифтами у CSS

Все багатство і розмаїтість існуючих шрифтів для української мови обмежено фактично трьома шрифтами: serif (звичайно Times або інший шрифт із засічками), sans-serif (Arial, Helvetica або інший шрифт без засічок) і monospace (Courier). Кожне з цих сімейств представлено лише одним кириличним шрифтом.

Для керування відображенням літер можливо застосувати кілька атрибутів, що впливають на шрифт:

- **font-family** – сімейство накреслень шрифту (гарнітура);
- **font-style** – пряме накреслення або курсив;
- **font-weight** – "посилення" (насиченість) шрифту, "жирність" букв;

▪ **font-size** - розмір шрифту (кегль). Задається в пікселях (px) і типографських пунктах (pt);

▪ **font-variant** - варіант накреслення (звичайний або дрібні букви - капітель).

Усі ці параметри можна сполучити в одному атрибуті font:

```
font:bold 12pt sans;
```

Специфікація CSS передбачає перерахування шрифтів в описах стилів, що дозволяє частково вирішити проблему підбора шрифту. На жаль, у Unix і Windows шрифти не погоджені. Фактично, при розробці сторінок у CSS використовуються лише класи шрифтів (serif, sans-serif і monospace).

Гарнітура (font-family)

Гарнітура шрифту - це набір накреслень одного шрифту. Шрифт може мати "пряме" накреслення (normal), курсив (italic), "скошене" (oblique), посилене по насиченості ("жирне", bold), "дрібне" (капітель, small-caps) і т.п.

Найбільш розповсюджені гарнітури в російській і українській частинах Web - це Times, Arial, Courier. Причому усі вони належать до різних груп шрифтів. Times - це пропорційний шрифт "із засічками" (serif), Arial - це пропорційний шрифт "без засічок" (sans-serif), а Courier - це моноширний шрифт (monospace). У Unix замість Arial частіше застосовується Helvetica.

У чому різниця між цими групами шрифтів, можна показати на прикладі (рис. 2.2):

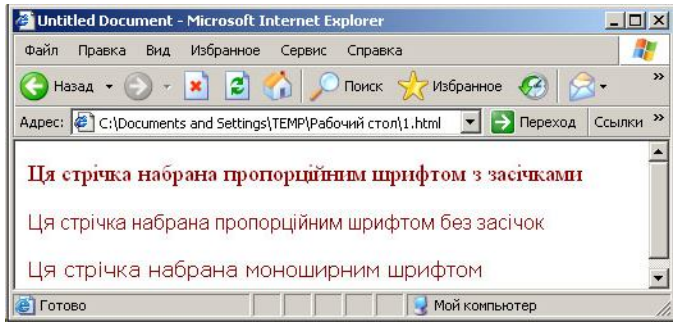


Рис. 2.2 Гарнітура шрифту

Для того, щоб скинути значення всіх змінних сесії, можна використовувати функцію session_unset(); Знищити поточну сесію цілком можна командою session_destroy(); Вона не скидає значення глобальних змінних сесії і не видаляє cookies, а знищує всі дані, асоційовані з поточною сесією.

```
<?
    session_start(); // ініціалізуємо сесію
    $test = "змінна сесії";
    $_SESSION['test']= $test;
    // реєструємо змінну $test.
    // якщо register_globals=on, то можна використовувати
    // session_register('test');
    print_r($_SESSION);
    // виводимо всі глобальні змінні
    echo session_id();
    // виводимо ідентифікатор сесії
    echo "<hr>";
    session_unset();
    // знищуємо всі глобальні змінні сесії
    print_r($_SESSION);
    echo session_id();
    echo "<hr>";
    session_destroy(); // знищуємо сесію
    print_r($_SESSION);
    echo session_id();
?>
```

У результаті роботи цього скрипта будуть виведені три рядки: у першому - масив з елементом test і його значенням, а також ідентифікатор сесії, у другому - порожній масив і ідентифікатор сесії, у третьому - порожній масив. Таким чином, видно, що після знищення сесії знищується і її ідентифікатор, і ми більше не можемо ні реєструвати змінні, ні взагалі робити які-небудь дії із сесією.

Хід виконання роботи

1. *Всі наступні завдання виконуйте у окремих каталогах на своєму ftp-акаунті, наприклад ...**/lr3/task1/***

2. *Завдання 1.* Створити на своєму ftp-акаунті php-сторінку (**get_send.php**) з формою, що має елементи типів **hidden**, **text**, **radio** (використайте також властивість **checked**), **password**, **textarea**, **checkbox**, **select** (випадаюче меню), **reset** і **submit**. Призначення цієї сторінки і назви всіх її полів придумати самостійно.

Доступ до таких змінних здійснюється за допомогою масиву $_SESSION['ім'я_змінної']. Якщо ж у налаштуваннях php вімкнена опція register_globals, то до сесійних змінних можна звертатися ще і як до звичайних змінних, наприклад так: $ім'я_змінної.

Якщо register_globals=off (відключені), то користуватися session_register() для реєстрації змінних переданих методами POST або GET, не можна, тобто це просто не працює. І взагалі, не рекомендується одночасно використовувати обидва методи реєстрації змінних, $_SESSION і session_register().

Приклад. Реєстрація змінних Зареєструємо логін і пароль, що вводяться користувачем на сторінці авторизації.

```
<?
      session_start();
         // створюємо нову сесію або відновлюємо поточну
      if (!isset($_GET['go'])){
       echo "<form>
         Login: <input type=text name=login>
         Password: <input type=password name=passwd>
         <input type=submit name=go value=Go>
       </form>";}
      else {   $_SESSION['login']=$_GET['login'];
        // реєструємо змінну login
        $_SESSION['passwd']=$_GET['passwd'];
        // реєструємо змінну passwd
      // тепер логін і пароль - глобальні змінні для цієї сесії
      if ($_GET['login']=="pit" && $_GET['passwd']=="123") {
            Header("Location: secret_info.php");
            // перенаправляємо на сторінку secret_info.php}
      else echo "Невірне введення,спробуйте ще раз<br>";
      }
      print_r($_SESSION);
         // виводимо всі змінні сесії
?>
```

Знищення змінних сесії

Функція session_unregister(ім'я_змінної) знищує глобальну перемінну з поточної сесії (тобто знищує її зі списку зареєстрованих змінних). Якщо реєстрація виконується за допомогою $_SESSION ($HTTP_SESSION_VARS для версії PHP 4.0.6 і більш ранніх), то використовують мовну конструкцію unset(). Вона не повертає ніякого значення, а просто знищує зазначені змінні.

Кегль (font-size)

Кегль - це, якщо говорити спрощено, розмір шрифту. CSS через параметр font-size дозволяє керувати розміром літер.

Розмір шрифту можна задавати в типографських пунктах (pt, 0,35 мм) або пікселях (px). При установці кегля варто пам'ятати, що font-size задає не висоту літери, а розмір "прямокутника" під букву, що більше самої букви.

От кілька прикладів використання font-size:

```
<P STYLE="font-size:12pt;">Кегль параграфа встановлений у 12 пунктів</P>
<P STYLE="font-size:12px;">Кегль параграфа встановлений у 12 пікселів</P>
<P STYLE="font-size:120%;">Кегль параграфа встановлений у 120% від розміру
букв параграф елемента, що охоплює,</P>
```

Крім відсотків, існує ще кілька умовних одиниць виміру кегля, які можна застосовувати в CSS:

```
<P STYLE="font-size:large;">Розмір кегля large</P>
<P STYLE="font-size:small;">Розмір кегля small</P>
<P STYLE="font-size:x-small;">Розмір кегля x-small</P>
<P STYLE="font-size:xx-small;">Розмір кегля xx-small</P>
```

Аналогічно x-small і xx-small, існують розміри x-large і xx-large. Крім того, є larger, smaller і medium.

Накреслення

У кожної гарнітури (font-family) існує кілька накреслень. Кожне з них визначається в CSS трьома параметрами стилю: font-style, font-variant, font-weight.

Атрибут стилю font-style визначає пряме накреслення (normal) і курсив:

```
<P STYLE="color:darkred; font-style:normal;">Пряме накреслення</P>
<P STYLE="color:darkred; font-style:italic;">Курсив</P>
```

Якщо хочеться підсилити насиченість ("жирність") шрифту, то в описі стилю вказують атрибут font-weight, що приймає значення normal або bold:

```
<P STYLE="color:darkred;font-style:italic;font-weight:bold;">Курсив</P>
```

Для якісного відображення дрібних літер у деяких гарнітурах присутній накреслення капітель. У CSS для використання капітелі зарезервований атрибут font-variant, що приймає значення normal і small-caps. На практиці застосування

font-variant проблематично через відсутність капітелі в стандартному наборі кириличних шрифтів.

Текст у CSS

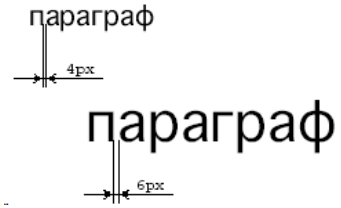
Розглянемо такі важливі характеристики текстового фрагменту, як:

- міжбуквенні відстані;
- висота рядків;
- вирівнювання;
- відступ у першому рядку параграфа;
- перетворення накреслення.

Усі ці атрибути згруповані у властивості текстових фрагментів (Text Properties).

Міжбуквенні відстані

Відстань між буквами автоматично регулюється розміром шрифту — кеглем. Чим більший розмір шрифту, тим більша відстань між буквами:



Моноширний шрифт обраний не випадково. На пропорційному шрифті міжбуквенна відстань залежить від накреслення букв і визначити її як відстань між буквами досить складно. У моноширинного шрифту розмір символу фіксований, тому і відстань між буквами простежується чітко.

Однак не завжди зручно керувати міжбуквенною відстанню через кегль (font-size). Бувають випадки, коли потрібно або ущільнити рядок, або збільшити відстані між буквами. Це можна зробити за допомогою атрибута letter-spacing:

```
<P STYLE="font-family:monospace;letter-spacing:5pt;color:black">
      Міжбуквенна відстань 5pt</P>
<P STYLE="font-family:monospace;letter-spacing:10pt;color:black">
      Міжбуквенна відстань 10pt</P>
```


Для наочності сесії можна задати ім'я за допомогою функції session_name([ім'я_сесії]). Робити це потрібно ще до ініціалізації сесії. Одержати ім'я поточної сесії можна за допомогою цієї ж функції, викликаної без параметрів: session_name();

Приклад. Створення сесії

Створимо сесію і подивимося, який вона одержить ідентифікатор і ім'я.

```
<? session_start();
      // створюємо нову сесію або відновлюємо поточну
   echo session_id();
   // виводимо ідентифікатор сесії
   ?>
   <html>
    <head><title>My home page</title></head>
   ... // домашня сторінка
   </html>
   <?
    echo session_name();
     // виводимо ім'я поточної сесії.
     // У даному випадку це PHPSESSID
?>
```

Реєстрація змінних сесії

Для того, щоб передавати і зберігати протягом сесії наші власні змінні (наприклад, логін і пароль) потрібно просто зареєструвати свої змінні:

```
session_register(ім'я_змінної1, ім'я_змінної2, ...);
```

Помітимо, що реєструються не значення, а імена змінних. Зареєструвати змінну досить один раз на будь-якій сторінці, де використовуються сесії. Імена змінних передаються функції session_register() без знака $. Усі зареєстровані в такий спосіб змінні стають глобальними (тобто доступними з будь-якої сторінки) протягом даної сесії роботи із сайтом.

Зареєструвати змінну також можна, просто записавши її значення в асоціативний масив $_SESSION, тобто написавши

```
$_SESSION['ім'я_змінної'] = 'значення_зміної';
```

У цьому масиві зберігаються всі зареєстровані (тобто глобальні) змінні сесії.

http://green.nsu.ru/test.php?PHPSESSID=ac4f4a45bdc893434c95dcaffb1c1811

Цей спосіб передачі ідентифікатора використовується автоматично, якщо в браузера, що відправив запит, вимкнені cookies. Він досить надійний – передавати параметри в командному рядку можна завжди. З іншого боку, ідентифікатор сесії можна підглянути, скористатися збереженим варіантом у рядку браузера або підробити. Хоча, звичайно, усі ці проблеми або надумані або їх можна розв'язати. Наприклад, хто зможе запам'ятати рядок з 32 різних символів? А якщо правильно організувати роботу із сесіями (вчасно їх знищувати), то навіть збережений у браузері номер сесії нічого не дасть.

Створення сесії

Перше, що потрібно зробити для роботи із сесіями, це запустити механізм сесій. Якщо в налаштуваннях сервера змінна session.auto_start встановлена в значення "0" (якщо session.auto_start=1, то сесії запускаються автоматично), то будь-який скрипт, у якому потрібно використовувати дані сесії, повинен починатися з команди

session_start();

Одержавши таку команду, сервер створює нову сесію або відновлює поточну, ґрунтуючись на ідентифікаторі сесії, переданому по запиті. Як це робиться? Інтерпретатор PHP шукає змінну, в якій зберігається ідентифікатор сесії (за замовчуванням це PHPSESSID) спочатку в cookies, потім у змінних, переданих за допомогою POST- і GET-запитів. Якщо ідентифікатор знайдений, то користувач вважається ідентифікованим, виконується зміна всіх URL і виставляння cookies. Інакше користувач вважається новим, для нього генерується новий унікальний ідентифікатор, потім виконується зміна URL і виставляння cookies.

Команду session_start() потрібно викликати у всіх скриптах, у яких має бути використані змінні сесії, причому до виведення яких-небудь даних у браузер. Це пов'язано з тим, що cookies виставляються тільки до виведення інформації на екран.

Одержати ідентифікатор поточної сесії можна за допомогою функції session_id().

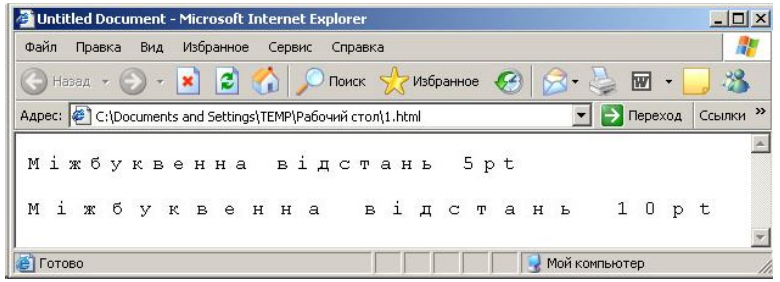


Рис. 2.3 Відстань між буквами

Вирівнювання

По замовчуванні усі слова в параграфі притиснуті вліво.

У звичайній HTML-розмітці такий ефект досягається за рахунок застосування атрибута ALIGN:

```
<P ALIGN=justify>...</P>
```

Аналогічний результат у CSS досягається за рахунок атрибута text-align:

```
<P STYLE="text-align:right; color:black;">
    Цей параграф вирівняний по правому краю. Усі рядки праворуч
    закінчуються на границі розділу. А от ліворуч вони починаються
    з різним відступом від лівого краю
</P>
```

Вирівнювати текст можна в будь-якому блоковому елементі. Центрувати текст:

```
<P STYLE="text-align:center;">...</P>
```

Перетворення шрифту

Перетворення шрифту має на увазі капіталізацію слів, заміну всіх "великих" і "маленьких" літер у великі, або, навпаки, одержання одних рядкових літер:

```
<P STYLE="text-transform:uppercase;">Зробити великими</P>
<P STYLE="text-transform:lowercase;">Зробити рядковими</P>
<P STYLE="text-transform:capitalize;">
        Зробити великими перші букви в словах</P>
```

Ще один вид перетворення шрифту – це підкреслення, перекреслення або надкреслення слів. Виконується таке перетворення за допомогою атрибуту text-decoration:

```
<P STYLE="text-decoration:line-through;">Перекреслимо це речення</P>
<P STYLE="text-decoration:underline;">Підкреслимо це речення</P>
```

Перший рядок параграфа

При оформленні параграфів у технології CSS можна скористатися абзацним відступом – атрибут text-indent.

Горизонтальний відступ у першому рядку параграфа щодо його лівого краю:

```
<P STYLE="text-indent:20pt;">Цей параграф ми почнемо з рядка
    з горизонтальним відступом у двадцять типографських пунктів від
    лівого краю параграфа.</P>
<P STYLE="text-indent:-10pt;">А в цьому параграфі ми застосуємо від'ємний
    горизонтальний відступ у першому рядку параграфа.</P>
```

Крім text-indent у CSS для оформлення першого рядка параграфа зарезервований модифікатор стилю first-line. Він дозволяє не лише задати горизонтальний зсув, але і визначити інші параметри параграфа:

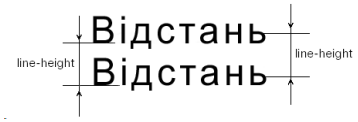
```
P:first-line { color:red; }
```

Ще один параметр, що впливає на відображення першого рядка параграфа – перша буква першого рядка. Її відображенням керує модифікатор first-letter:

```
P:first-letter { font-size:20pt; }
```

Міжстрічкова відстань

У CSS міжстрічкова відстань визначається параметром line-height. Він задає відстань не між рядками, а між базовими лініями рядків. Простіше кажучи, якщо, наприклад, взяти букву "н" і надрукувати її послідовно одну під одною, то line-height буде дорівнює відстані між двома однаковими точками букв



Подивимося, як цей параметр впливає на взаємне розташування рядків:

```
<P STYLE="line-height:12pt;font-size:12pt;color:black;">
        Цей параграф ми набрали кеглем 12 pt.Line-height заданий у
12 pt.</p>
<P STYLE="line-height:24pt;font-size:12pt;color:black;">
```


QUERY_STRING – інформація, що знаходиться в URL після знака питання;

SCRIPT_NAME – віртуальний шлях до програми, що повинна виконуватися;

HTTP_USER_AGENT – інформація про браузер, що використовує клієнт

Механізм сесій

Сесії – це механізм, що дозволяє створювати і використовувати змінні, що зберігають своє значення протягом усього часу роботи користувача із сайтом.

При цьому щораз, заходячи на сайт, користувач одержує нові значення змінних, що дозволяють ідентифікувати його протягом цього сеансу або сесії роботи із сайтом. Звідси і назва механізму - сесії.

Задача ідентифікації користувача розв'язується шляхом присвоєння кожному користувачеві унікального номера, так званого ідентифікатора сесії (SID, Session IDentifier). Він генерується PHP у той момент, коли користувач заходить на сайт, і знищується, коли користувач іде із сайта, і являє собою рядок з 32 символів наприклад,

ac4f4a45bdc893434c95dcaffb1c1811

Цей ідентифікатор передається на сервер разом з кожним запитом клієнта і повертається назад разом з відповіддю сервера.

Існує кілька способів передачі ідентифікатора сесії:

За допомогою cookies. Cookies були створені спеціально як метод однозначної ідентифікації клієнтів і являють собою розширення протоколу HTTP. У цьому випадку ідентифікатор сесії зберігається в тимчасовому файлі на комп'ютері клієнта, що послав запит. Метод, безсумнівно, гарний, але багато користувачів відключають підтримку cookies на своєму комп'ютері через проблеми з безпекою.

За допомогою параметрів командного рядка. У цьому випадку ідентифікатор сесії автоматично вбудовується в усі запити (URL), передані серверові, і зберігається на стороні сервера.

Наприклад: адреса

http://green.nsu.ru/test.php перетворюється на адресу

Незважаючи на всі ці недоліки, використовувати метод GET досить зручно при налаштуванні скриптів (тоді можна бачити значення й імена переданих змінних) і для передачі параметрів, що не впливають на безпеку.

Для методу POST

Вміст форми кодується точно так само, як для методу GET, але замість додавання рядка до URL вміст запиту посилається блоком даних як частина операції POST. Якщо є присутнім атрибут ACTION, то значення URL, що там знаходиться, визначає, куди посилати цей блок даних. Цей метод, як вже відзначалося, рекомендується для передачі великих по обсязі блоків даних.

Інформація, введена користувачем і відправлена серверові за допомогою методу POST, подається на стандартне введення програмі, зазначеної в атрибуті action, або поточному скрипту, якщо цей атрибут опущений. Довжина файлу, що посилається, передається в змінній CONTENT_LENGTH, а тип даних - у змінній CONTENT_TYPE.

Передати дані методом POST можна тільки за допомогою HTML-форми, оскільки дані передаються в тілі запиту, а не в заголовку, як у GET. Відповідно і змінити значення параметрів можна, тільки змінивши значення, введене у форму. При використанні POST користувач не бачить передані серверові дані.

При відправленні даних на сервер будь-яким методом передаються не тільки самі дані, введені користувачем, але і ряд змінних, названих змінними оточення, що характеризують клієнта, історію його роботи, шляхи до файлів і т.п. Нижче наведені деякі змінні оточення:

REMOTE_ADDR – IP-адреса хоста (комп'ютера), що відправляє запит;

REMOTE_HOST – ім'я хоста, з якого відправлений запит;

HTTP_REFERER – адреса сторінки, що посилається на поточний скрипт;

REQUEST_METHOD – метод, що був використаний при відправленні запиту;


```
Цей параграф ми набрали кеглем 12 pt.Line-height заданий у 24 pt.</P>
    <P STYLE="line-height:6pt;font-size:12pt;color:black;">
Цей параграф ми набрали кеглем 12 pt.Line-height заданий у 6 pt.</P>
```

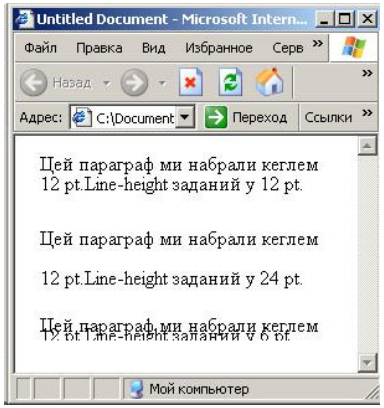


Рис. 2.4 Міжстрічкова відстань

Перший приклад набраний зі значенням line-height, рівним розмірові кегля. В другому прикладі значення line-height удвічі перевищує розмір кегля. У третьому прикладі значення line-height у два рази менше розміру кегля – рядка стали "наповзати" один на одного.

Списки у CSS

При відображенні списків CSS дозволяє керувати формою і зображенням "кульок" (bullets) списку і заміняти "кульки" картинками.

Керування відображенням елементів списку віднесено до набору властивостей, у які входить атрибут display. У цього атрибута може бути лише одне значення – none, що означає він не відображається браузером взагалі:

```
<UL STYLE="display:none;">
    <LI>Перший елемент списку
    <LI>Другий елемент списку
    <LI>Третій елемент списку
</UL>
```

Форма "кульок"

Часто в якості "кульки" застосовують квадрат або інший символ типографського набору, а також графічну картинку.

CSS дозволяє керувати формою "кульки" через атрибут list-style-type:

```
<UL STYLE="list-style-type:square;">
<LI>У вигляді "кульки" використовуємо квадрат
</UL>
<UL STYLE="list-style-type:disk;">
<LI>У вигляді "кульки" використовуємо диск
</UL>
<UL STYLE="list-style-type:circle;">
<LI>У вигляді "кульки" використовуємо коло
</UL>
```

У впорядкованих списках (OL):

```
<OL STYLE="list-style-type:lower-roman; color:black;">
<LI> У вигляді "кульки" стрічкові римські літери
</OL>
<OL STYLE="list-style-type:upper-alpha; color:black;">
<LI> У вигляді "кульки" великі латинські літери
</OL>
<OL STYLE="list-style-type:lower-alpha; color:black;">
<LI> У вигляді "кульки" стрічкові латинські літери
</OL>
```

CSS дозволяє взагалі відмовитися від "кульок". Для цього потрібно вказати значення атрибута list-style-type рівним none.

"Кульки"-картинки

У такої "кульки" є URL, що використовується в CSS для звертання до неї.

```
<UL STYLE="list-style-image:url(barrow.gif);">
        <LI>Елемент списку розташований за стрілкою
</UL>
```

Позиціонування

Координати і розміри

Стандарт CSS-P дозволяє з точністю до пікселя розмістити блоковий елемент розмітки в робочому полі вікна браузера. CSS-P визначає дві системи координат: відносну й абсолютну.

Блоки - це не абстрактні точки, що не займають на площині сторінки місця. Блоки представляють собою прямокутники, що "замітають" площу. Текст і інші компоненти сторінки під блоком стають недоступні користувачеві, тому

Тут action – це URL-адреса програми, що повинна обробляти форму (це або програма, задана в атрибуті action тега form, або сама поточна програма, якщо цей атрибут опущений). Імена name1, name2, name3 відповідають іменам елементів форми, а value1, value2, value3 – значенням цих елементів. Усі спеціальні символи, включаючи = і &, в іменах або значеннях цих параметрів будуть опущені. Тому не варто використовувати в назвах або значеннях елементів форми ці символи і символи кирилиці в ідентифікаторах.

Якщо в поле для введення ввести який-небудь службовий символ, то він буде переданий у його шіснадцятковому коді, наприклад, символ $ заміниться на %24. Так само передаються і кириличні літери.

Для полів введення тексту і пароля (це елементи input з атрибутом type=text і type=password), значенням буде те, що введе користувач. Якщо користувач нічого не вводить у таке поле, то в рядку запиту буде присутній елемент name=, де name відповідає імені цього елемента форми.

Для кнопок типу checkbox і radio button значення value визначається атрибутом VALUE у тому випадку, коли кнопка відзначена. Не відзначені кнопки при складанні рядка запиту ігноруються цілком. Кілька кнопок типу checkbox можуть мати один атрибут NAME (і різні VALUE), якщо це необхідно. Кнопки типу radio button призначені для одного з усіх запропонованих варіантів і тому повинні мати однаковий атрибут NAME і різні атрибути VALUE.

У принципі створювати HTML-форму для передачі даних методом GET не обов'язково. Можна просто додати в рядок URL потрібні змінні і їхні значення.

```
http://phpbook.info/test.php?id=10&user=pit
```

У зв'язку з цим в передачі даних методом GET є один істотний недолік – кожен може підробити значення параметрів. Тому не радимо використовувати цей метод для доступу до захищеним паролем сторінок, для передачі інформації, що впливає на безпеку роботи програми або сервера. Крім того, не варто застосовувати метод GET для передачі інформації, що не дозволено змінювати користувачеві.

ЛАБОРАТОРНА РОБОТА № 3

Тема: Дослідження методів передачі даних між WEB-сторінками GET, POST, SESSIONS.

Мета: Дослідження особливостей методів GET, POST, SESSIONS і набуття навичок у застосуванні їх у WEB-сторінках.

Теоретичні відомості

Методи передачі даних

Будь-який запит клієнта до сервера повинен починатися з вказівки методу. Метод повідомляє про мету запиту клієнта. Протокол HTTP підтримує досить багато методів, але реально використовуються тільки три:

GET
HEAD
POST

Використання HTML-форм для передачі даних на сервер

Як передавати дані серверові? Для цього в мові HTML є спеціальна конструкція – форми. Форми призначені для того, щоб отримувати від користувача інформацію. Отже, для створення форми в мові HTML використовується тег FORM. У середині нього знаходиться одна або кілька команд INPUT. За допомогою атрибутів action і method тега FORM задаються ім'я програми, що буде обробляти дані форми, і метод запиту, відповідно. Команда INPUT визначає тип і різні характеристики запитуваної інформації. Відправлення даних форми відбувається після натискання кнопки input типу submit.

Отже, у формі можна вказувати метод передачі даних. Подивимося, що буде відбуватися, якщо вказати метод GET або POST, і в чому буде різниця.

Для методу GET

При відправленні даних форми за допомогою методу GET вміст форми додається до URL після знака питання у вигляді пар **ім'я =значення, об'єднаних за допомогою амперсанта &:**

```
action?name1=value1&name2=value2&name3=value3
```


лінійні розміри блоку мають для створення HTML-сторінок не менше значення, ніж його координати.

Абсолютні координати

При використанні "абсолютних" координат точка відліку міститься у верхньому лівому куті вікна браузера, а осі X і Y спрямовані вправо по горизонталі і вниз по вертикалі, відповідно.

Якщо в цій системі координат деякий блоковий елемент повинен бути розміщений на 10 px нижче верхнього краю робочої області браузера і на 20 px правіше лівого краю робочої області браузера, то його опис буде виглядати наступним чином:

```
.example { position:absolute;top:10px;left:20px; }
```

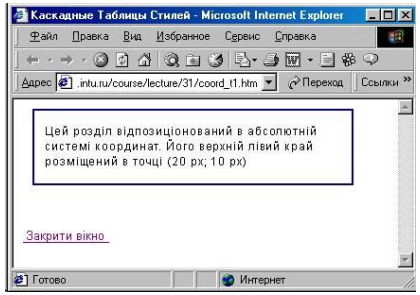


Рис. 2.5 Блоковий в абсолютній системі координат

У даному записі тип системи координат заданий атрибутом position (значення - absolute), координата X задана атрибутом left (значення - 20 px), координата Y - атрибутом top (значення - 10 px).

Атрибути top і left визначають координати верхнього лівого кута блоку в абсолютній системі координат.

Значення координат можуть бути і від'ємними. Для того, щоб забрати з відображуваної області блок з лінійними розмірами 100 px (висота) на 200 px (ширина), досить позиціонувати його в такий спосіб:

```
.example{position:absolute;top:-100px;left:-200px;width:200px;height:100px;}
```

Абсолютне позиціонування застосовується тоді, коли або весь зміст сторінки повинен бути доступним без скролінга ("прокручування"), або коли елементи розмітки знаходяться на

початку сторінки і їхнє взаємне розташування важливе з погляду дизайну, наприклад, для використання випливаючих меню.

Відносні координати

Дана координатна система дозволяє розмістити блоки на сторінці в координатах їх охоплюючого блоку, що дозволяє зберігати взаємне розташування елементів розмітки при будь-якому розмірі вікна браузера і його налаштувань по замовчуванні.

Як початок відліку в цій системі координат обрана точка розміщення поточного блоку по замовчуванні. Вісь X при цьому спрямована горизонтально вправо, а вісь Y - вертикально вниз.

```
<DIV STYLE="border-width:1px;border-style:solid;width:100%;height:100px;">
    <DIV STYLE="position:relative;top:0px;left:0px;border-width:1px;">
        Цей блок знаходиться в точці відліку відносних координат
    </DIV>
        <DIV STYLE="position:relative; top:0px;left:50px; border-width:1px;">
            А цей блок зміщений вправо на 50px
        </DIV>
</DIV>
```

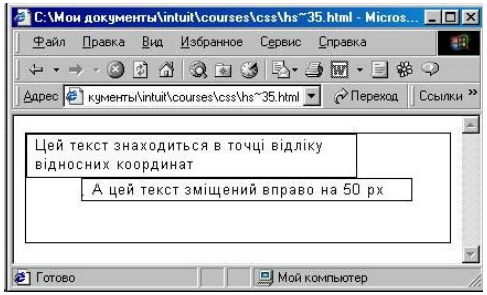


Рис. 2.6 Блок заданий відносними координатами

Для роботи з відносною системою координат краще користуватися універсальними блоками DIV. У відносній системі координат можна користуватися від'ємними зміщеннями:

```
<DIV STYLE="position:relative;top:0;left:50;border-width:1px;
                    border-style:solid;width:200px;">
<A HREF="javascript:if(flag==0)
{window.document.layers[2].left=-50;flag=1;}
else
{window.document.layers[2].left=50;flag=0;};
void(0);">
Змістити шар
```


Контрольні запитання:

1.Наведіть властивості *шрифтів і фону* і приклади CSS-коду для їх управління.

2.Наведіть властивості *тексту* і приклади CSS-коду для їх управління.

3.Наведіть властивості *гіперпосилань* і приклади CSS-коду для їх управління.

4.Наведіть властивості *таблиць* і приклади CSS-коду для їх управління.

5.Наведіть властивості *списків* і приклади CSS-коду для їх управління.

6.Поясніть принцип роботи меню, що побудовано на властивостях списків.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>hover</title>
  <style type="text/css">
   ul {
    width: 180px; /* Ширина меню */
    list-style: none; /* Для списку прибираємо маркери */
    margin: 0; /* І немає відступів навколо */
    padding: 0; /* Прибираємо поля навколо тексту */
    font-family: Arial, sans-serif; /* Рублений шрифт для тексту меню */
    font-size: 10pt; /* Розмір назв в пункті меню */
   }
   li ul {
    position: absolute; /* Підменю має абсолютну позицію */
    display: none; /* Ховаємо підменю */
    margin-left: 165px; /* Зсуваємо підменю вправо */
    margin-top: -2em; /* Зсуваємо підменю вверх */
   }
   li a {
    display: block; /* Посилання як блоковий елемент */
    padding: 5px; /* Поля навколо надпису */
    text-decoration: none; /* Прибираємо підкреслення для гіперпосилань */
    color: #666; /* Колір тексту */
    border: 1px solid #ccc;     /* Рамка навколо пунктів меню */
    background-color: #f0f0f0; /* Колір фону */
    border-bottom: none; /* Прибираємо границю знизу */
   }
   li a:hover {
    color: #ffe; /* Колір текста активного пункту */
    background-color: #5488af; /* Колір фону активного пункту */
   }
   li:hover ul {
    display: block; /* При виділенні пункту курсором міші відтворюється підменю */
   }
   .brd {
    border-bottom: 1px solid #ccc; /* Лінія знизу */
   }
  </style>
 </head>
 <body>

  <ul id="menu">
   <li><a href="ukr.html">Українські страви</a>
    <ul>
     <li><a href="u1.html">Борщ</a></li>
     <li><a href="u2.html">Галушки</a></li>
     <li><a href="u3.html">Вареники</a></li>
     <li><a href="u4.html" class="brd">Котлети по-київськи</a></li>
    </ul>
   </li>
   <li><a href="by.html">Російські страви</a>
    <ul>
     <li><a href="b.html">Щі</a></li>
     <li><a href="b2.html">Жаркое</a></li>
     <li><a href="b3.html">Компот</a></li>
     <li><a href="b4.html" class="brd">Суп з грибів</a></li>
    </ul>
   </li>
   <li><a href="ge.html">Грузинські страви</a>
    <ul>
     <li><a href="g1.html">Суп-харчо</a></li>
     <li><a href="g2.html">Хачапурі</a></li>
     <li><a href="g3.html">Чихіртма</a></li>
     <li><a href="g4.html" class="brd">Шашлик</a></li>
    </ul>
   </li>
  </ul>

 </body>
</html>
```



```
    </A>
    </DIV>
```

У даному прикладі шар, спочатку зміщений на 50 пікселів вправо, після натискання на гіпертекстове посилання зміщається на 100 пікселів вліво, одержуючи від'ємну величину зсуву по осі X (left:-50 px). Після повторного натискання на посилання положення блоку відновлюється.

Лінійні розміри блоку

Лінійні розміри блоку задаються двома параметрами: шириною (width) і висотою (height) блоку.

```
<P STYLE="width:200px;height:100px;background-color:black;color:white;">
    <SPAN STYLE="color:white;">
    ...
    </SPAN>
</P>
```

Хід виконання роботи

1.Завантажити операційну систему Linux.

2.Запустити файловий менеджер (ФМ) Krusader і приєднатись по **ftp** до сервера, використовуючи логін і пароль свого власного **ftp**-облікового запису. В з'єднанні використати сервер **192.168.1.5** або **vsau.vin.ua** (Нагадаємо, що для вмикання **ftp**-з'єднання в файловому менеджері Krusader треба або натиснути Ctrl_N або ж скористатись його меню). Відкрийте у WEB-браузері і в редакторі Krusader файл **1.html** (див. лаб. роботу №1) і продовжуйте досліджувати властивості стилів, виконуючи наступни пункти.

3. *Дослідження властивостей шрифтів і фону.* Послідовно додайте до стилю **h1** властивості **font-size, font-family** і т.д. і кожну з них разом з відповідним рисунком занотуйте у своєму блозі (дослідіть також і закоментовані опції):

```
h1 {
        color: red;
        background: #f0f0f0;
        font-style: italic; /*oblique*/
        font-weight: normal; /*bold normal*/
        font-variant: small-caps; /*normal*/
        font-size: 10pt;
        font-height: 10.2em;
        font-family: monospace; /*sans-serif serif fantasy*/
        font-size: 10pt;
```

```
}
```

4. *Дослідження властивостей тексту.* В будь-якому місці блоку **<body>...</body>** додайте шар **t1** з таким текстом:

```
<div id='t1'>
        Властивості ТЕКСТУ.<br>
        Рядок1.<br>
        Рядок2.<br>
</div>
```

і до нього пропишіть секцію у стилях для дослідження властивостей тексту:

```
#t1 {
        text-align: none; /* justify left right */
        text-decoration: blink; /* line-through overline underline blink
        none */
        text-indent: 1.5em; /*  */
    }
```

Послідовно додайте до стилю #t1 властивості **font-size, font-family** і т.д. і кожну з них разом з відповідним рисунком занотуйте у своєму блозі (дослідіть також і закоментовані опції).

12. *Дослідження властивостей гіперпосилань.* В будь-якому місці блоку **<body>...</body>** додайте такий **html**-код:

```
<br>
    <a href='http://vsau.vin.ua'>Інтранет-сайт ВНАУ</a>
    <a href='abc.html'>Невідвідуване. Натиснути сюди тільки після
    завершення досліджень і не раніше!</a>
<br>
```

і до нього пропишіть секцію у стилях для дослідження властивостей тексту:

```
a { /* тут пропишемо властивості для всіх станів гіперпосилань */
  /* поля навколо гіперпосилань */
 }
a:visited {
        }
a:hover  {
        }
```

Додайте до стилів **a** властивості, як би надали гіперпосиланню таку поведінку:

- границі навколо гіперпосилання складають 10px;
- відвідуване гіперпосилання має зелений колір, сірий фон і його текст перекреслений;

- при наведенні курсору на нього текст стає підкресленим, білим, 18 кеглю, на синьому фоні.

Занотуйте у своєму блозі спостереження разом з рисунками.

6. *Дослідження властивостей таблиць.* Відкрийте "Генератор стилів HTML and CSS для таблиць" і, натискаючи кнопки, уважно дослідіть властивості таблиць. Після натискання кнопок відслідковуйте зміни **HTML**-коду і **CSS** у вікні нижче. Занотуйте властивості табличних **CSS** до своїх блогів.

7. *Дослідження розташування тексту.* В будь-якому місці блоку **<body>...</body>** додайте такий **html**-код:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>

<div style="font-family: Times, serif; font-size: 200%">
        T<span style="vertical-align: sub">E</span>X и L<span
        style="vertical-align: 5px; font-size: 80%">A</span>T<span
        style="vertical-align: sub">E</span>X
</div>

</body>
</html>
```

Перегляньте його в WEB-браузері і опишіть у блозі принцип його роботи.

8. *Використання властивостей списків для створення меню.* Створіть файл **menu.html** на своєму **ftp**-акаунті і занесіть **html**-код, що наведений нижче. Дослідіть його роботу. Опишіть принцип роботи меню у блозі.

9. Створіть файл **my_menu.html** і, користуючись розумінням принципу побудови, створіть ГОРИНТАЛЬНЕ меню до свого проекта з літньої практики по курсу ТСПІС.