

**ВНЗ „Львівський кооперативний коледж економіки і права”**

(повна назва навчального закладу)

Циклова (предметна) комісія **інформатики та розробки програмного  
забезпечення**

***WEB-ДИЗАЙН***

***МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ТА ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ  
РОБОТИ СТУДЕНТІВ КООПЕРАТИВНИХ НАВЧАЛЬНИХ ЗАКЛАДІВ  
ДЕННОЇ ФОРМИ НАВЧАННЯ***

***спеціальність 5.05010301 «Розробка програмного забезпечення»***

**РОЗРОБЛЕНО ВИКЛАДАЧЕМ:  
ВАРГАС В.М.**

Розглянуто на засіданні циклової комісії інформатики

та комп'ютерної техніки

Протокол №1 від \_\_\_\_\_ 200\_\_р.

Голова циклової комісії:

\_\_\_\_\_

**Львів – 2016**

## Пояснювальна записка

Самостійна робота студента (СРС) – це форма організації навчального процесу, за якої заплановані завдання виконуються студентом самостійно під методичним керівництвом викладача.

Мета СРС – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

У ході самостійної роботи студент має перетворитися на активного учасника навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними і практичними знаннями, вільно орієнтуватися в інформаційному просторі, нести індивідуальну відповідальність за якість власної професійної підготовки.

СРС включає: опрацювання лекційного матеріалу; опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами дисципліни; підготовку до лабораторних занять; поглиблене опрацювання окремих лекційних тем або питань; виконання індивідуальних завдань (вирішення розрахункових індивідуальних та комплексних завдань) за вивченою темою; пошук (підбір) та огляд літературних джерел за заданою проблематикою дисципліни; аналітичний розгляд наукової публікації; контрольну перевірку студентами особистих знань за запитаннями для самодіагностики; підготовку до контрольних робіт та інших форм поточного контролю; підготовку до модульного контролю; систематизацію вивченого матеріалу з метою підготовки до семестрових заліків та екзамену.

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною технічною літературою в області веб-програмування та веб-дизайну. Основні види самостійної роботи, які запропоновані студентам для засвоєння теоретичних знань з навчальної дисципліни, наведені в табл. 1.

## Самостійна робота

№ з/п	Назва теми	Форма контролю	Література	Кількість годин
<i>1-й семестр</i>				
	<b>Модуль 1 Модуль 1. Основи web</b>			
1	<b>Тема 1. Структура і принципи Веб. Уведення в HTML</b> <i>Універсальний ідентифікатор – URI. Схеми адресації ресурсів в Internet. Принципи роботи HTTP. Особливості HTML Використання шарів в HTML. Підходи щодо оптимізації HTML-сторінок</i>	Захист індивідуального завдання	Основна [1,3,4,5,7,8] Додаткова [15,16,17,18,19]	10
	<b>Модуль 2. Технологія CSS</b>			
2	<b>Тема 2. Технологія CSS та її підтримка браузерами</b> <i>Створення меню, що випадає. CSS 3: елементи із згладженими кутами; лінійні і сферичні градієнти; додавання до елементів і до тексту елементів тіні; небезпечні шрифти; анімація і різні ефекти переходів</i>	Захист індивідуального завдання	Основна [1,3,4,5] Додаткова [15]	8
3	<b>Тема 3. Блокова верстка сторінок веб-сайта</b> <i>Коласуючу відступи, потік документу, накладення і порядок шарів, плаваючі елементи, позиціонування елементів</i>	Захист індивідуального завдання	Основна [3,4,5,6,10,11] Додаткова [16,17,18]	7
4	<b>Тема 4. Основи JavaScript</b> <i>Динамічні форми. Читання та запис строк у файли cookie. Перетворення символів строки в Unicode та у зворотному напрямку. Кодування та декодування URL. Обробка помилок сценарію.</i>	Захист індивідуального завдання	Основна [6,7] Додаткова [15,19]	8
<b>Всього</b>				<b>33</b>

Тема лекції 2. Структура і принципи Веб. Уведення в HTML 5

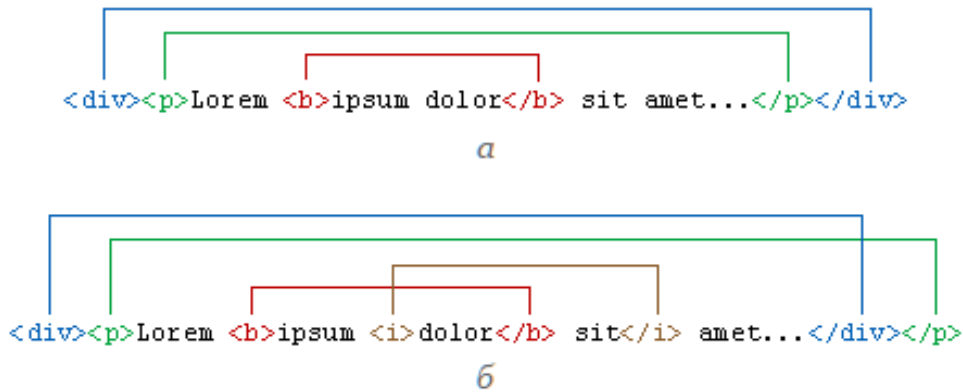


Рис.1.

Вкладеність тегів, а- правильне, б- неправильне

Теги HTML, призначені для виділення фрагментів тексту

Тег	Призначення	Як відображає браузер
<ABBR>	Абревіатура	Підкресленим
<ACRONYM >	Абревіатура	Фактично те ж саме, що і тег <ABBR>
<CITE>	Невелика цитата	Курсивом
<DEL>	Текст, віддалений з Web-сторінки	Зачеркнутим
<DFN>	Новий термін	Курсивом
<EM>	Менш важливий текст	Курсивом
<INS>	Текст, знову поміщений на Web-сторінку	Підкресленим
<KBD>	Дані, що вводяться користувачем у яку-небудь програму	Моноширинним шрифтом
<Q>	Невелика цитата.	Фактично те ж саме, що і тег <CITE>
<SAMP>	Дані, що виводяться якою-небудь програмою	Моноширинний шрифтом
<STRONG>	Дуже важливий текст	Напівжирним шрифтом
<TT>	Дані, що виводяться якою-небудь програмою.	Фактично те ж саме, що і тег <SAMP>
<VAR>	Ім'я змінної у вихідному коді програми	Курсивом

Вставка неприпустимих символів. Літерали

Недопустимі символи	Літерали
-(Довге тире)	& mdash;
- (Коротке тире)	& nda
"	& amp;
<	& lt;
>	& gt;
©	& copy;
®	& reg;
Символ євро	& euro;

## типи MIME

Формат файлів	Тип MIME
WAV	audio/wave audio/wav audio/x-wav audio/x-pn-wav
OGG	audio/ogg (для аудіофайлів) video/ogg (для відеофайлів) application/ogg
MP4	video/mp4
QuickTime	video/quicktime

### **Тема лекції 3. Технологія CSS та її підтримка браузером** **Властивості форматування елементів засобами CSS**

#### **1) Параметри шрифту:**

##### **1.1.) Атрибут стилю font-family задає ім'я шрифту**

Особливе значення **inherit** вказує, що текст даного елемента Web-сторінки має бути набраний тим же шрифтом, що і текст батьківського елемента. Кажуть, що в даному випадку елемент Web-сторінки "успадковує" шрифт від батьківського елемента. Це, до речі, значення атрибута стилю font-family за замовчуванням

##### **1.2.) Атрибут стилю font-size визначає розмір шрифту:**

font-size: <розмір> | xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | inherit

Розмір шрифту можна задати в абсолютних і відносних величинах.

#### **Одиниці виміру розміру, підтримувані стандартом CSS**

Назва	Відображення в CSS
пікселі	px
пункти	pt
дюйми	in
сантиметри	cm
міліметри	mm
піки	pc
Розмір букви "m" поточного шрифту	em
Розмір букви "x" поточного шрифту	ex
Відсотки від розміру шрифту батьківського елемента	%

### 1.3) Атрибут стилю **color** задає колір тексту:

color: <колір> | inherit

1.4) Атрибут стилю **opacity** дозволяє вказати ступінь напівпрозорості елемента Web-сторінки:

opacity: <числове значення> | inherit

### 1.5) Атрибут стилю **font-weight** встановлює "жирність" шрифту:

font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

### 1.6) Атрибут **font-style** задає накреслення шрифту:

font-style: normal | italic | oblique | inherit

1.7) Атрибут стилю **text-decoration** задає "прикраса" (підкреслення або закреслення), яке буде застосоване до тексту:

text-decoration: none | underline | overline | line-through | blink | inherit

1.8) Атрибут стилю **font-variant** дозволяє вказати, як будуть виглядати малі літери шрифту:

font-variant: normal | small-caps | inherit

### 1.9) Атрибут стилю **text-transform** дозволяє змінити регістр символів тексту:

text-transform: capitalize | uppercase | lowercase | none | inherit

### 1.10) Атрибут стилю **line-height** задає висоту рядка тексту:

line-height: normal | <відстань> | inherit

1.11) Атрибут стилю **letter-spacing** дозволяє задати додаткову відстань між символами тексту:

letter-spacing: normal | <відстань>

1.12) атрибут стилю **font**, що дозволяє задати одночасно відразу декілька параметрів шрифту:

font: [<накреслення>] [<вид малих літер>] [<"жирність">] [<розмір> [/ <висота рядка тексту>]] <ім'я шрифту>

Наприклад, для заголовків шостого рівня - шрифт Arial розміром 12 пунктів і курсиву:  
H6 {font: italic 12pt Verdana}

## 2) Параметри, керуючі розривом рядків

2.1.) Атрибут стилю **white-space** задає правила, якими Web-браузеру керується при виведенні тексту. Зокрема, з його допомогою ми можемо змінити правила, встановлені за замовчуванням і . Формат запису цього атрибута стилю:

**white-space: normal | pre | nowrap | pre-wrap | pre-line | inherit**

2.2.) Атрибут стилю **word-wrap** застосовується нечасто, але в деяких випадках без нього не обійтися. Він дозволяє вказати місця, в яких Web-оглядач може виконати розрив тексту:

**word-wrap: normal | break-word | inherit**

## 3) Параметри вертикального вирівнювання

Атрибут стилю **vertical-align** якраз задає вертикальне вирівнювання тексту:

vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | <проміжок між базовими лініями> | inherit

## 4) Параметри тіні у тексту

Параметри тіні задає атрибут стилю **text-shadow**:

text-shadow: none | <колір> <горизонтальне зміщення> <вертикальне зміщення> [<радіус розмиття>]

## 2) Установки фону

### 2.1. Атрибут стилю **background-color** служить для завдання кольору фону:

`background-color: transparent | <колір> | inherit`

Значення `transparent` прибирає фон зовсім; тоді елемент Web-сторінки отримає "прозорий" фон.

**2.2. За допомогою атрибуту стилю `background-position` можна вказати позицію фонового зображення щодо елемента Web-сторінки, для якого воно призначено:**

`background-position: <горизонтальна позиція> [<вертикальна позиція>] | inherit;`

Горизонтальна позиція фонового зображення задається в наступному форматі:

`<числове значення> | left | center | right`

**2.3. Коли ми прокручуємо вміст Web-сторінки у вікні Web-обозревателя, разом з нею прокручується і фонове зображення (якщо воно є). Стандарт CSS пропонує забавну можливість - заборона прокрутки графічного фону Web-сторінки і фіксація його на місці. Фіксацією фону управляє атрибут стилю **background-attachment**:**

`background-attachment: scroll | fixed;`

Значення `scroll` змушує Web-оглядач прокручувати фон разом з вмістом Web-сторінки (це поведінка за замовчуванням). Значення `fixed` фіксує фон на місці, і він не буде прокручуватися.

## 3) Параметри абзаців, списків і відображення

### 3.1. Параметри тексту

**Атрибут стилю `text-align` задає горизонтальне вирівнювання тексту:**

`text-align: left | right | center | justify | inherit`

### 3.2. Параметри списків

**Атрибут стилю `list-style-type` задає вид маркерів або нумерації у пунктів списку:**

`list-style-type: disc|circle|square|decimal|decimal-leading-zero|lower-roman|upper-roman|lower-greek|lower-alpha|lower-latin|upper-alpha|upper-latin|armenian|georgian|none|inherit`

**Атрибут стилю `list-style-image` дозволяє задати в якості маркера пунктів списку якесь графічне зображення (створити графічний маркер).**

`list-style-image: none | <інтернет-адресу файлу зображення> | inherit`

**Атрибут стилю `list-style-position` дозволяє вказати місце розташування маркера або нумерації в пункті списку:**

`list-style-position: inside | outside | inherit`

Доступних значень тут два:

- `Inside` - маркер або нумерація знаходяться як би всередині пункту списку, є його частиною;

- `Outside` - маркер або нумерація винесені за межі пункту списку (це звичайна поведінка).

### 3.3. Параметри відображення

**Атрибут стилю `visibility` дозволяє вказати, чи буде елемент відображатися на Web-сторінці:**

visibility: visible | hidden | collapse | inherit

Атрибут стилю visibility застосовується досить рідко і лише для створення спеціальних ефектів, на зразок зникаючого або з'являються елементи Web-сторінки

**Атрибут стилю display** вельми примітний. Він дозволяє задати вигляд елемента Web-сторінки: буде він блоковим, вбудованим або взагалі пунктом списку.

Приклад:

display: none|inline|block|inline-block|list-item|run-in|table|inline-table|table-caption|table-column|table-column-group|table-header-group|table-row-group|table-footer-group|table-row|table-cell|inherit

У більшості практичних випадків достатньо значень none, block і inline атрибуту стилю display. Останні значення занадто специфічні, щоб часто їх застосовувати.

#### 4) Стандарт CSS пропонує засоби для створення відступів двох видів

**4.1. Атрибути стилю padding-left, padding-top, padding-right і padding-bottom** дозволяють задати величини **внутрішніх відступів**, відповідно, ліворуч, зверху, праворуч і знизу елемента Web-сторінки:

padding-left | padding-top | padding-right | padding-bottom: <відступ> | auto | inherit

Ми можемо вказати як величини відступу абсолютне або відносне значення. Значення auto задає величину відступу за умовчанням, зазвичай воно дорівнює нулю.

**4.2. Атрибути стилю margin-left, margin-top, margin-right і margin-bottom** дозволяють задати величини **зовнішніх відступів**, відповідно, ліворуч, зверху, праворуч і знизу:

margin-left | margin-top | margin-right | margin-bottom: <відступ> | auto | inherit

Тут також доступні абсолютні та відносні значення. Значення auto задає величину відступу за замовчуванням, як правило, рівне нулю

Зовнішні відступи ми також можемо вказати за допомогою атрибуту стилю **margin**. Він задає величини відступу одночасно з усіх боків елемента Web-сторінки:

margin: <відступ 1> [<відступ 2> [<відступ 3> [<відступ 4>]]]

Цей атрибут стилю поводиться так само, як його "колега" padding.

#### 5) Параметри рамки

CSS також дозволяє нам зробити суцільну, пунктирну або точкову рамку по уявній кордоні елемента Web-сторінки.

**Атрибути стилю border-left-width, border-top-width, border-right-width і border-bottom-width** задають товщину, відповідно, лівою, верхньої, правої і нижньої сторін рамки:

border-left-width | border-top-width | border-right-width | border-bottom-width: thin | medium | thick | <товщина> | inherit

Ми можемо вказати або абсолютне або відносне числове значення товщини рамки, або одне із зумовлених значень: thin (тонка), medium (середня) або thick (товста). В останньому випадку реальна товщина рамки залежить від Web-оглядача. Значення товщини за замовчуванням також залежить від Web-оглядача, тому її завжди краще встановлювати явно.

**Атрибут стилю border-width** дозволяє вказати значення товщини відразу для всіх сторін рамки: border-width: <товщина 1> [<товщина 2> [<товщина 3> [<товщина 4>]]]

**Атрибути стилю border-left-color, border-top-color, border-right-color і border-bottom-color** задають колір, відповідно, лівою, верхньої, правої і нижньої сторін рамки:

border-left-color | border-top-color | border-right-color | border-bottom-color: transparent | <колір> |



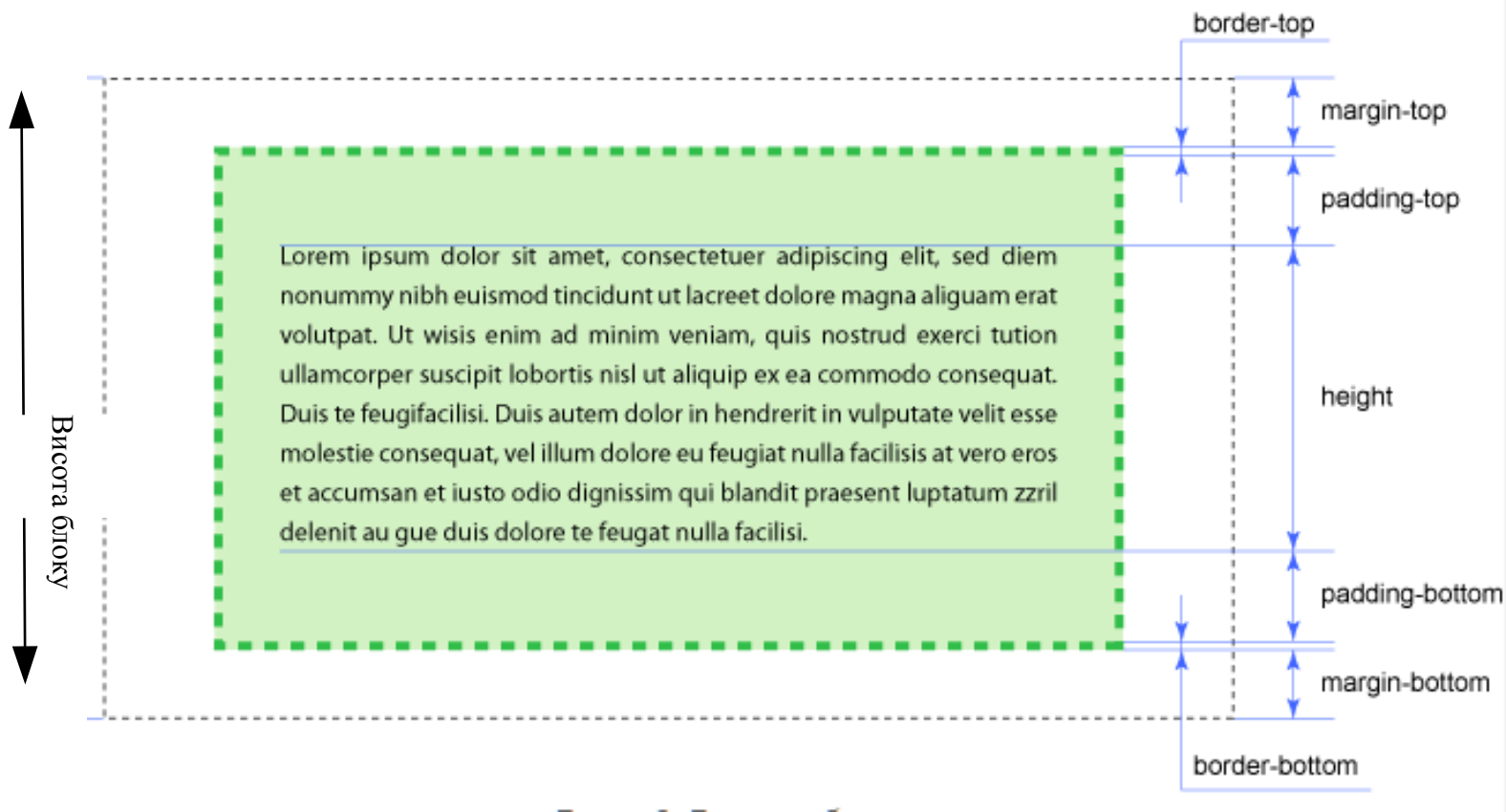
inherit

Значення transparent задає "прозорий" колір, крізь який буде "просвічувати" фон батьківського елемента.

#### Тема лекції 4. Блокова верстка сторінок веб-сайта.



**Рис.2 Ширина блока**



***Рис.3. Висота блоку***

## Позиціонування елементів

**Позиціонуванням** називається положення елемента в системі координат. Розрізняють чотири типи позиціонування: нормальне, абсолютне, фіксоване і відносне. Залежно від типу, який встановлюється через властивість `position`, змінюється і система координат.

Завдяки комбінації властивостей `position`, `left`, `top`, `right` і `bottom` елемент можна накладати один на інший, виводити в точці з певними координатами, фіксувати у вказаному місці, визначити положення одного елемента щодо іншого та ін. Подібно іншим властивостям CSS управління позиціонуванням доступно через скрипти. Таким чином, можна динамічно змінювати положення елементів без перезавантаження сторінки, створюючи анімацію і різні ефекти.

### Нормальне позиціонування

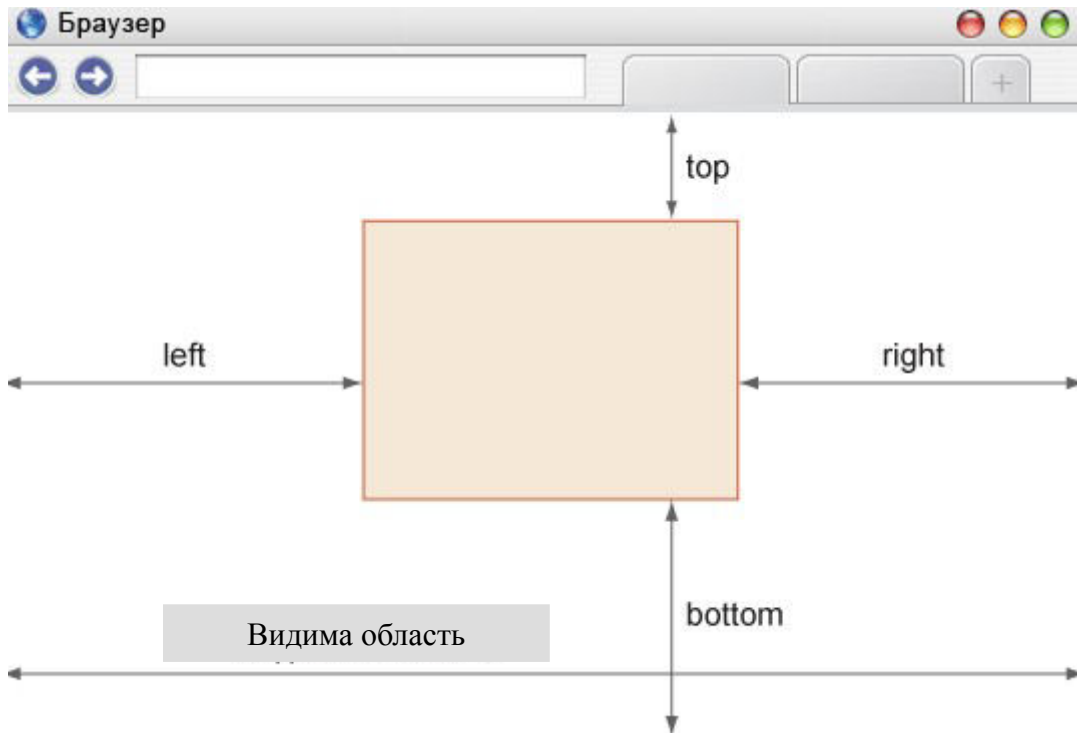
Якщо для елемента властивість `position` не задано або його значення `static`, елемент виводиться в потоці документа як звичайно. Іншими словами, елементи відображаються на сторінці в тому порядку, як вони йдуть у вихідному коді HTML.

Властивості `left`, `top`, `right`, `bottom` якщо визначені, ігноруються.

### Абсолютне позиціонування

При абсолютному позиціонуванні елемент не існує в потоці документа і його положення задається щодо країв браузера. Задати цей тип можна через значення `absolute` властивості `position`. Координати вказуються щодо країв вікна браузера, званого «видимою областю» (рис.4).

Рис.4.



Значення властивостей left, top, right, bottom

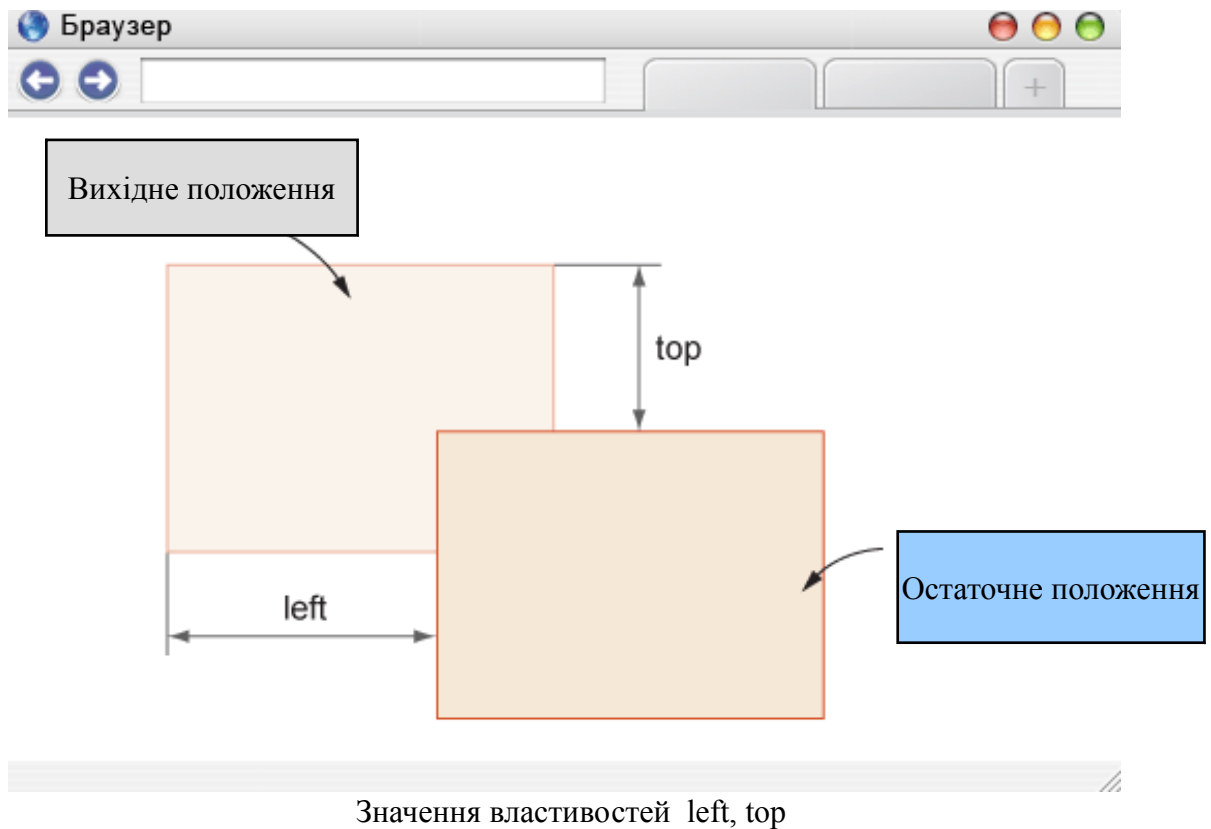
### фіксоване положення

Фіксоване положення шару задається значенням fixed властивості position і по своїй дії схоже на абсолютне позиціонування. Але на відміну від нього прив'язується до зазначеної властивостями left, top, right і bottom точки на екрані і не міняє свого положення при прокручуванні веб-сторінки. Ще одна різниця від absolute полягає в тому, що при виході фіксованого шару за межі видимої області справа або знизу від неї, не виникає смуг прокрутки.

### відносне позиціонування

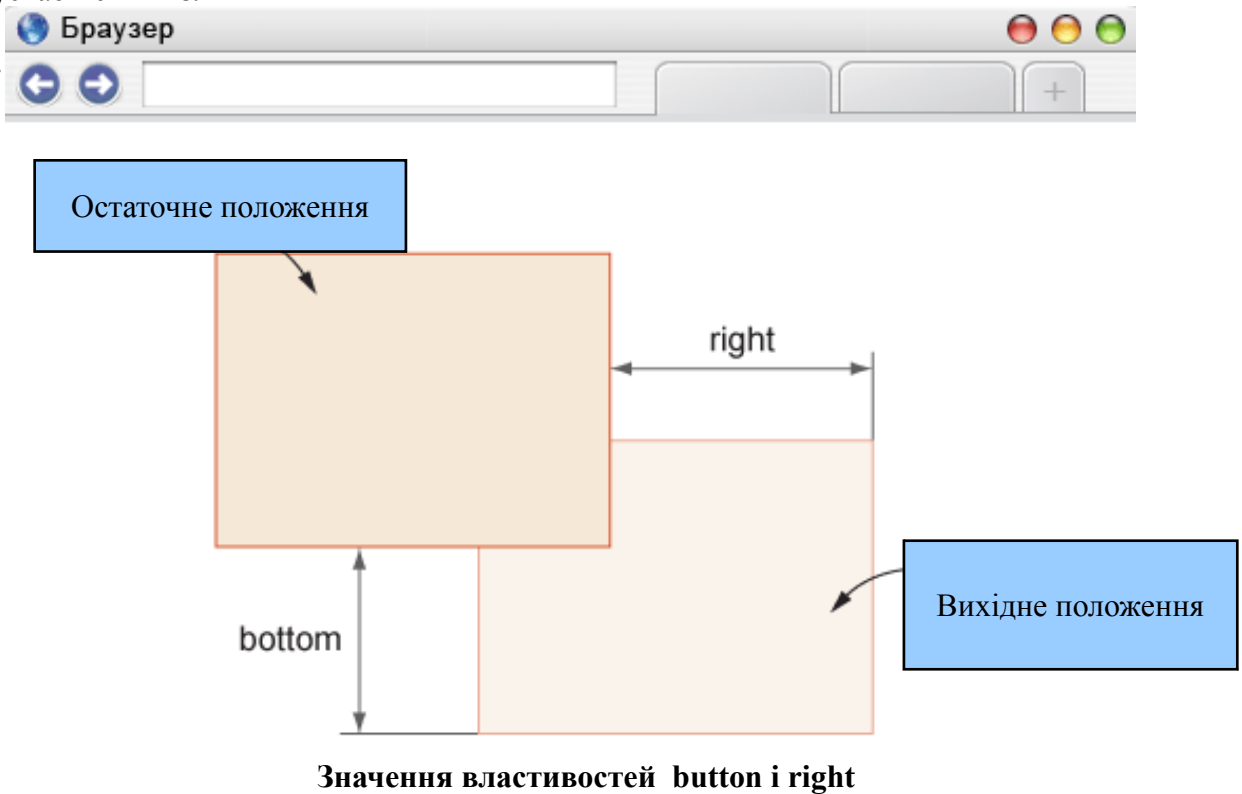
Якщо задати значення relative властивості position, то положення елемента встановлюється щодо його вихідного місця. Додавання властивостей left, top, right і bottom змінює позицію елемента і зрушує його в ту або іншу сторону від первісного розташування. Позитивне значення left визначає зсув вправо від лівої межі елемента, негативне - зрушення вліво. Позитивне значення top задає зсув елемента вниз (рис. 5), негативне - зрушення вгору.

Рис.5.



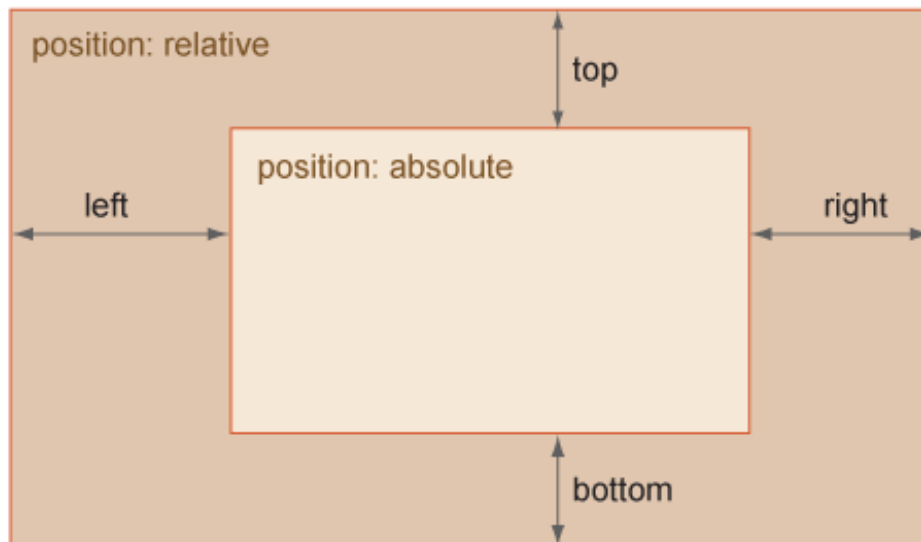
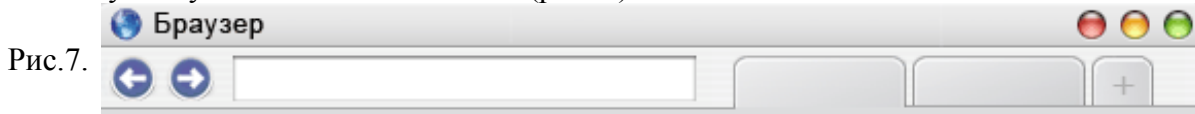
Властивості `button` і `right` виробляють зворотний ефект. При позитивному значенні `right` зрушує елемент вліво від його правого краю, при негативному - зрушує вправо (рис. 6). При позитивному значенні `bottom` елемент піднімається вгору, при негативному опускається вниз.

Рис.6.



вкладені шари

Зазвичай відносно позиціонування саме по собі застосовується не часто, оскільки є ряд властивостей виконують фактично ту ж роль, наприклад, той же `margin`. Але поєднання різних типів позиціонування для вкладених шарів є одним із зручних і практичних прийомів верстки. Якщо для батьківського елемента задати `relative`, а для дочірнього `absolute`, то відбудеться зміна системи координат і положення дочірнього елемента при цьому вказується щодо його батька (рис. 7).



Значення властивостей **left**, **top**, **right**, **bottom** у вложенних слоях

## Стили, що задають параметри блоків

### 1) Параметри розмірів

**Атрибути стилю першої групи задають розміри блоків.** Власне, їх можна застосовувати не тільки в контейнерах, а й в будь-яких інших блочних елементах.

**Атрибути стилю `width` і `height`** дозволяють задати, відповідно, ширину і висоту елемента Web-сторінки:

`width: auto | <ширина> | inherit` `height: auto | <висота> | inherit`

Ми можемо вказати абсолютне значення розміру, скориставшись будь-який з доступних в CSS одиниць виміру. Тоді розмір елемента буде незмінним:

```
#navbar {width: 100px}
```

Можна задати відносне значення розміру у відсотках.

### 2) Параметри розміщення

**Атрибут стилю `float`** якраз і задає, по якого краю батьківського елемента буде вирівнюватися даний елемент Web-сторінки:

**float: left | right | none | inherit**

**Атрибут стилю `clear`** надає можливість однозначно вказати, що даний блоковий елемент в будь-якому випадку повинен розташовуватися нижче плаваючих контейнерів, що передують йому.: **clear: left | right | both | none | inherit**

*Питання для самоконтролю*

Тема Способи додавання стилів на сторінку

## Питання для перевірки

1. Сайт має більше ста HTML-документів, що мають однакове стильове оформлення. Який спосіб підключення CSS підходить найкраще?

- Пов'язані стилі.
- Глобальні стилі.
- Блокові стилі.
- Внутрішні стилі.
- Експорт стилю.

2. У даному прикладі, який колір буде у тексту на веб-сторінці?

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset = "utf-8">
  <title> Колір тексту </ title>
  <style>
    HTML {color: black; }
    BODY {color: red; }
    P {color: green; }
  </ style>
</ head>
<body>
  <p style = "color: blue;"> <span style = "color:
olive;"> Текст </ span> </ p>
</ body>
</ html>
```

- Подивитися приклад
- Подивитися приклад
- Чорний
- Червоний
- Зелений
- Синій
- Оливковий

3. Який HTML-код застосовується для підключення зовнішнього CSS-файлу?

```
<style> mystyle.css </ style>
<style> @ mystyle.css </ style>
<link rel = "stylesheet" href = "http://htmlbook.ru/mystyle.css">
<link>import url (mystyle.css) </ link>
<stylesheet> mystyle.css </ stylesheet>
```

4. Який атрибут використовується для визначення внутрішнього стилю?

- style
- class
- styles
- font
- link

### Відповіді

1. Пов'язані стилі.
2. Оливковий.
3. <link rel = "stylesheet" href = "http://htmlbook.ru/mystyle.css">
4. style

## Перелік типових питань на залік з дисципліни

### *WEB-ДИЗАЙН*

1. Охарактеризуйте сучасний web-дизайн. Концепцію 2.0.
2. Охарактеризуйте сучасний стан мови HTML.
3. На які секції поділена Web-сторінка.
4. Кодування Unicode, що це таке.
5. Атрибути HTML-тегів
6. Структурування тексту в документі HTML.
7. Які бувають списки у HTML.
8. Що таке *Текст фіксованого формату*? Яким тегом він позначається у документі HTML.
9. Перелічіть і охарактеризуйте кілька тегів, які відповідають за форматування документа у HTML.
10. Як здійснюється *вставка неприпустимих символів* у документі HTML.
11. Вставка графічних зображень у документі HTML.
12. Що таке форми і як вони створюються у документі HTML.
13. Як відображається мультимедіа у документі HTML.
14. Як теги відповідають за створення таблиць у документі HTML.
15. Засоби навігації у документі HTML (*гіперпосилання*)
16. Види стилів CSS. Охарактеризуйте один.
17. Спеціальний селектор у CSS.
18. Що таке таблиця стилів.
19. Види псевдоелементів у CSS.
20. Псевдокласи у CSS.
21. Коментарі CSS
22. Які ви знаєте види блоків у CSS.
23. Що ви знаєте про старі різновиди Web-дизайну та їхню критику.
24. Охарактеризуйте блокову модель.
25. Верстка веб-сторінок, що це таке.
26. Які ви знаєте макети по ширині.
27. переваги і недоліки фіксованого макету.
28. Переваги і недоліки гумового макету.
29. Охарактеризуйте макет по колонках.
30. Що таке web- сценарії.

## Рекомендована література

### Основна

1. Будилов В. А. Основы программирования для Интернета / В. А. Будилов. – СПб. : БХВ-Петербург, 2003. – 736 с.
2. Глушков С. В. Программирование Web-страниц / С. В. Глушков, И. А. Жакин, Т. С. Хачиров. – Х. : Фолио, 2005. – 390 с.
3. Калиновский А. И. Юзабилити: как сделать сайт удобным / А. И. Калиновский. – Мн. : "Новое знание", 2005. – 220 с.
4. Кристофер Шмитт CSS. Рецепты программирования / Кристофер Шмитт. – СПб. : "БХВ-Петербург", 2007. – С. 592.
5. Лещев Д. Создание интерактивного web-сайта : учебный курс / Д. Лещев. – СПб. : Питер, 2003. – 544 с.
6. Ломов А. Ю. HTML, CSS, скрипты: практика создания сайтов / А. Ю. Ломов. – СПб. : БХВ-Петербург, 2006. – 416 с.
7. Пономаренко В. С. Основы технологий Internet : учебное пособие / В. С. Пономаренко, С. В. Минухин, И. А. Торохтий. – Х. : Изд. ХГЭУ, 2001. – 256 с.
8. Прохоренок Н. HTML, Javascript, PHP и MySQL. Джентльменский набор Web-мастера / Н. Прохоренок. – СПб. : БХВ-Петербург, 2008. – 640 с.
9. Бен Фрейн. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. — СПб.: Питер, 2014. — 304 с.
10. Создание Web-страниц и Web-сайтов. Самоучитель / под ред. В. Н. Печникова. – М. : Изд-во Триумф, 2006. — 464 с.
11. Соколов С. А. HTML и CSS в примерах, типовых решениях и задачах. Профессиональная работа / С. А. Соколов. – М. : "Вильямс", 2007. – 416 с.
12. Эрик А. Мейер CSS-каскадные таблицы стилей: подробное руководство / А. Эрик Мейер. – М. : "Символ", 2006. — С. 576.
13. Microsoft ASP.NET 2.0. Базовый курс. Мастер-класс / пер. с англ. – М. : Издательство "Русская Редакция" ; СПб. : Питер, 2007. – 688 с. 11.2.

### Додаткова

14. Валентайн Ч. XHTML / Ч. Валентайн, К. Минник. – М. : Издательский дом "Вильямс", 2001. – 480 с.
15. Дарнелл Р. JavaScript: справочник / Р. Дарнелл. – СПб. : "Питер", 2000. – 192 с.
16. Зельдман Д. Web-дизайн по стандартам / Д. Зельдман. – СПб. : Изд-во: НТ Пресс, 2005. – 440 с.
17. Кирсанов Д. Веб-дизайн / Д. Кирсанов. – СПб. : "Символ- Плюс", 1999. – 376 с.
18. Круг С. Веб-дизайн / С. Круг. – СПб. : "Символ-Плюс", 2001. – 195 с.



19. Моррисон М. HTML и XML. Быстро и эффективно / М. Моррисон. – СПб. : "Питер", 2005. – 302 с.
20. Нильсен Я. Веб-дизайн / Я. Нильсен. – СПб. : "Символ- Плюс", 2000. – 512 с.
21. Хольцшлаг М. Э. Использование HTML и XHTML / М. Э. Хольцшлаг. – М. : ВИЛЬЯМС, 2003. – 728 с.
22. Довідкове керівництво по ASP.NET [Електронний ресурс]. – Режим доступу : <http://msdn.microsoft.com/ruru/asp.net/default.aspx>.
23. Довідник по Web-мовам [Електронний ресурс]. – Режим доступу : [www.spravkaweb.ru/](http://www.spravkaweb.ru/).
24. Підручник PHP [Електронний ресурс]. – Режим доступу : [php.rus-phpnuke.com/](http://php.rus-phpnuke.com/).
25. Підручник по PHP 4 [Електронний ресурс]. – Режим доступу : [www.softtime.ru/bookphp/gl1\\_1.php](http://www.softtime.ru/bookphp/gl1_1.php).
26. Сайт ASP.NET. – Режим доступу : <http://www.asp.net/>.
27. Сайт компанії Microsoft. – Режим доступу : [www.microsoft.com](http://www.microsoft.com).
28. HTML довідник. – Режим доступу : [html.manual.ru/](http://html.manual.ru/).