

Веб-технології

ЛАБОРАТОРНА РОБОТА №7

Тема: Контейнери. дизайн з використанням CSS3. Шрифти, тіні, градієнти.

Мета: Навчитись застосовувати блокові контейнери на практиці різні типи стильового оформлення сторінок.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Контейнер – елемент Web-сторінки, призначений тільки для виділення фрагменту елемента. Це може бути частина елемента, весь елемент або навіть декілька блокових елементів.

Контейнери є **блокові та вбудовані**.

Розглянемо **вбудовані**. Вони є частиною блокового елемента сторінки. Створюються за допомогою парного тега ****. Фрагмент поміщають безпосередньо в цей тег.

```
01 <p> тут почався абзац з <span> дуже важливою </span> інформацією. </p>
```

До цього контейнера можна прив'язати якийсь стиль:

```
01 bolded { font-weight: bold }
02 <p> тут почався абзац з <span class="bolded"> дуже важливою </span>
03 інформацією. </p>
```

Розглянемо **блокові контейнери**. Вони можуть зберігати тільки блокові елементи (як один, так і декілька): абзаци, заголовки, великі цитати, теги адреси, текст фіксованого формату, таблиці, аудіо та відеоролики. Блоковий контейнер може містити як один блоковий елемент, так і декілька. Блокові контейнери формуються з допомогою парного тега **<DIV>**, всередині якого поміщають код, що формує вміст контейнера, наприклад:

```
01 <div>
02   <table border=2>
03     <thead><td>&nbsp;
04     <tbody>
05       <tr>
06         <td>стпівка 1 </td>
07         <td> комірка 1.2 </td>
08       </tr>
09       <tr>
10         <td>стпівка 2</td>
11         <td> комірка 2.2</td>
12       </tr>
13     </tbody>
14   </table>
15 </div>
```

```
01 <div style="text-align: center">
```

```
02 <video src="film.ogg" controls>
03 </video>
04 <p>для перегляду - натисніть кнопку відтворення</p>
05 </div>
```

Зазвичай окремі контейнери створюються для заголовку сайту, полоси навігації, основного змісту та відомостей про авторські права. Якщо основний зміст має складну структуру, то і для кожної з частин створюють окремий контейнер.

Для того, щоб задати різні параметри блокових контейнерів передбачені спеціальні атрибути стилів CSS. Це ширина і висота контейнера, місце розміщення та їх поведінка при переповненні.

Якщо написано прості теги, то власне кажучи, нічого не відбудеться і блоки розмістяться один під одним. Для розміщення блоків у потрібних місцях потрібно підключити файл зі стилями. Це робиться наступним чином:

```
01 <link rel="stylesheet" type="text/css" href="шлях до файлу зі стилями">
```

Після цього все слід поєднувати до потрібних блоків.

```
01 <div id="назва ідентифікатора">
02     <!--контент блоку -->
03 </div>
```

Почергово це відбувається приблизно так. Спочатку створюються 2 файли: Index.html – з розміткою сторінки, і Style.css – файл зі стилями.

Приклад розмітки:

```
01 <!doctype html>
02 <html lang="en">
03 <head>
04     <meta charset="UTF-8">
05     <title>Software News</title>
06     <link rel="stylesheet" type="text/css" href="style.css"> </head>
07 <body>
08     <div id="header"><!--тут підключення стилю для шапки сайту-->
09         <h2 align="center">Шапка (header)</h2>
10     </div>
11     <div id="menu"><!-- тут підключення стилю для меню сайту-->
12         Ліва колонка (left)
13     </div>
14     <div id="content"><!-- тут підключення стилю для основної області
15 сайту-->
16         Вміст сторінки (content)
17     </div>
18     <div id="footer"><!--тут підключення стилю для підвалу сайту-->
19         <h4 align="center">Підвал (footer)</h4>
20     </div>
21 </body>
22 </html>
```

Приклад сторінки зі стилями

```
01 body, html {
02     width: 100%;
```

```
03         padding:0;
04         margin:0;
05     }
06     #header{
07         border: 1px solid black;
08         background-color: #FDBFF4;
09     }
10     #menu{
11         width: 20%;
12         border: 1px solid black;
13         float: left;
14         background-color: #BFFDFA;
15     }
16     #content{
17         width: 79%;
18         border: 1px solid black;
19         float: right;
20         background-color: #FBF8B9;
21     }
22     #footer{
23         width: 100%;
24         height: 50px;
25         overflow: hidden;
26         text-align: center;
27         padding:0;
28         margin:0;
29     }
30 }
31 #footer h4{
32     background-color: #BFFDC7;
33     border: 1px solid black;
34     padding:0;
35     margin:0;
36 }
```

CSS3 надає можливість застосовувати наступні елементи стильового оформлення до веб-сторінок:

- створювати елементи із згладженими кутами;
- створювати лінійні та сферичні градієнти;
- більш гнучко оформляти фонову картинку елементів;
- створювати тіні до елементів та тексту;
- використовувати нестандартні шрифти;
- створювати анімацію і різні ефекти переходів та ін.

Створення елементів із згладженими кутами

За допомогою нової CSS3 властивості **border-radius** можна робити кути елементів згладженими.

Приклад використання:

```
01 #element1 {
02     border-radius: 10px;
03 }
```

Така властивість може застосовуватися не до всіх кутів елемента, а тільки до певних:

- **border-top-left-radius** робить згладженим тільки верхній лівий кут елемента;

- **border-top-right-radius** робить згладженим тільки верхній правий кут елемента;
- **border-bottom-left-radius** робить згладженим тільки нижній лівий кут елемента;
- **border-bottom-right-radius** робить згладженим тільки нижній правий кут елемента.

Додавання тіней до тексту

За допомогою нової CSS3 властивості **text-shadow** можна додавати до тексту елементів тіні (до тексту одного елемента може бути додано одночасно декілька тіней).

При заданні тіні для тексту необхідно вказати: величину зміщення тіні від тексту по горизонталі і вертикалі (може бути негативною), а також радіус розмиття і колір тіні.

Приклад використання

```
01 #shadow1 {  
02     text-shadow: 3px 2px #FFAE00;  
03 }  
04 #shadow2 {  
05     text-shadow: 1px 1px 10px #FFAE00;  
06 }  
07 #shadow3 {  
08     text-shadow: 2px 2px 2px #FFAE00, 2px 2px 15px #1435AD;  
09 }
```

Додавання до елементів тіні

За допомогою властивості **box-shadow** можна додавати до елементів сторінки тіні.

Тінь може бути зовнішньою та внутрішньою. Зовнішні тіні створюють ефект випуклості елемента над іншим вмістом, а внутрішні створюють ефект вдавненості елемента.

Приклад використання:

```
01 #element1 {  
02     box-shadow: 4px 4px black;  
03 }  
04 #element1 {  
05     box-shadow: 0px 0px 6px 2px black inset;  
06 }
```

Встановлення зображень як границі блоку

У CSS3 було додано нову властивість **border-image**, яка дозволяє вставляти довільні зображення як межу елемента.

Для того, щоб зробити це, необхідно виконати наступні дії:

- вказати шлях до зображення-границі;
- вказати величину відступу від кожного краю зображення для того, щоб розрізати зображення на 9 частин (верхній лівий кут, верхня сторона, верхній правий кут, ліва сторона і т.д.);
- вказати ширину границі в пікселях;
- вказати чи повинно зображення повторюватися (**repeat**), округлятися (**round**) або розтягуватися (**stretch**), щоб заповнити границю елемента.

Приклад використання

```
01 #element1 {  
02     border-image: url("images/imgborder.jpg") 30 30 round;  
03 }
```

Створення прозорих елементів та картинок

Для створення прозорих елементів у всіх браузерах використовується властивість **opacity**: **x**, де **x** — значення, яке може змінюватися від 0.0 (повністю прозорий елемент) до 1.0 (повністю непрозорий елемент).

Приклад використання

```
01 #opacity1 {  
02     opacity: 0.8;  
03 }
```

Лінійні градієнти

В CSS3 є вбудовані властивості для створення градієнтів. Так як в CSS3 сам браузер відмальовує градієнти, необхідність додаткових запитів градієнтних картинок у сервера відпадає і це дозволяє збільшити швидкість завантаження сторінок.

Лінійні градієнти створюються за допомогою CSS3 методу **linear-gradient**, який повинен вказуватися в значення властивості **background**.

Для того, щоб створити лінійний градієнт необхідно вказати його напрямок (може задаватися за допомогою ключових слів або градусів) і кольору переходу.

Приклад використання

```
01 #lineargradient1 {  
02     background: linear-gradient(top,white,black);  
03     background: -webkit-linear-gradient(top,white,black); /*для Chrome і  
04     Safari*/  
05     background: -o-linear-gradient(top,white,black); /*для Opera*/  
06     background: -moz-linear-gradient(top,white,black); /*для Firefox */  
07     background: -ms-linear-gradient(top,white,black); /*для IE10+*/  
08 }
```

Кольори переходу — це кольори, які приймає градієнт в певних його точках, наприклад градієнт, який плавно змінює колір з білого на чорний має білий колір переходу в початковій точці і чорний в кінцевій.

Лінійні градієнти можуть мати необмежену кількість кольорів переходу.

Можна вказувати координати розташування кольорів за допомогою % (0% — початок градієнта, 100% — кінець).

```
01 #lineargradient1 {  
02     background: linear-gradient(left,white 0%,green 50%,black 100%);  
03 }
```

Сферичні градієнти

За допомогою методу **radial-gradient** можна створювати сферичні градієнти.

Синтаксис визначення сферичних градієнтів дуже схожий на синтаксис лінійних, але вимагає також завдання форми градієнта (може бути сферичної або еліпсоїдної).

Приклад використання

```
01 #gradient1 {  
02     background: radial-gradient(white 20%,black 40%);  
03 }  
04 #gradient2 {  
05     background: radial-gradient(circle,#8F04A8 0%,#5D016D 40%,black 60%);  
06 }
```

Гرادієнти, що повторюються

Повторювані градієнти задаються за допомогою CSS3 методів **repeating-linear-gradient** (створює повторюваний лінійний градієнт) і **repeating-radial-gradient** (створює повторюваний сферичний градієнт).

Для того, щоб створити повторюваний градієнт необхідно вказати напрямок градієнта, а також колір переходу і відстань, яку вони повинні займати.

Приклад використання

```
01 #gradient1 {  
02     background:repeating-linear-gradient(50deg,white,white      5px,black  
03     5px,black 10px);  
04 }  
05 #gradient2 {  
06     background:repeating-radial-gradient(circle,#8F04A8      0%,#5D016D  
07     40%,black 60%);  
08 }
```

CSS3 властивість @font-face

У попередніх версіях CSS розробники були змушені використовувати тільки ті шрифти, які гарантовано встановлені на комп'ютері користувача, в CSS3 розробники можуть використовувати будь шрифти, які вони захочуть.

Коли є необхідний шрифт достатньо просто розмістити його на веб-сервері і підключити його за допомогою нової CSS3 властивості @font-face.

Підключений шрифт буде завантажено і встановлено на комп'ютер користувача автоматично при перегляді сторінки.

Приклад використання

```
01 @font-face {  
02     font-family: Consolas; /* Задаємо ім'я шрифту */  
03     src: url('Consolas.ttf'), /* Вказуємо його місцезнаходження у форматі  
04     .ttf */  
05     url('Consolas.eot') /* Вказуємо його місцезнаходження у форматі .eot  
06     */;  
07 }
```

Для того, щоб потім використовувати підключений шрифт необхідно просто додати до бажаного елемента властивість **font-family**, що містить ім'я цього шрифту.

```
01 div {  
02     font-family: Consolas;  
03 }
```

Приклад сторінки зі стилями

```
01 body, html {  
02     width: 100%;  
03     padding:0;  
04     margin:0;  
05 }  
06 #header{  
07     border: 1px solid black;  
08     background-color: #FDBFF4;  
09 }  
10 #menu{
```

```
11         width: 20%;
12         border: 1px solid black;
13         float: left;
14         background-color: #BFFDFA;
15     }
16     #content{
17         width: 79%;
18         border: 1px solid black;
19         float: right;
20         background-color: #FBF8B9;
21     }
22     #footer{
23         width: 100%;
24         height: 50px;
25         overflow: hidden;
26         text-align: center;
27         padding:0;
28         margin:0;
29     }
30     #footer h4{
31         background-color: #BFFDC7;
32         border: 1px solid black;
33         padding:0;
34         margin:0;
35     }
```

Завдання для роботи

1. Передбачте у оформленні сторінки використання окремих стилів для абзаців основної частини, списків (наприклад, меню), хоча б однієї таблиці.
2. Додайте до сайту елементи дизайну з використанням CSS3 (блоки із заокругленими кутами, градієнти, шрифти, тіні та ін.).
3. Оформіть ваші контейнери різними стилями, використовуючи Додаток до лабораторної роботи, передбачивши відступи та рамки.
4. Представте роботу викладачеві.
5. Зробіть висновок про роботу.

Контрольні питання

1. Що таке контейнер? Які є види контейнерів? Опишіть їх.
2. Які переваги та недоліки контейнера?
3. Перерахуйте параметри контейнерів. Які атрибути їх описують?
4. Що таке переповнення контейнера? Які можливості для його усунення? (знайти самостійно і коротко описати у звіті)
5. Які можливості стильового оформлення сторінки засобами CSS3?
6. Опишіть порядок створення елементів зі згладженими кутами.
7. Опишіть порядок додавання тінювого ефекту до тексту та елементів сторінки.
8. Опишіть порядок встановлення зображення як границі блоку.
9. Опишіть порядок створення прозорих елементів та картинок.
10. Які типи градієнтів можна оформити засобами CSS3?
11. Як організувати підключення та використання довільного шрифту?

Додаток до ЛР №7

Деякі атрибути CSS та їх можливі значення

Атрибут	Значення	Пояснення
Параметри шрифту		
font-family	<Список імен шрифтів через кому> inherit	Ім'я шрифту
font-size	<розмір> xx-small x-small small medium large x-large xx-large larger smaller inherit	Розмір шрифту
color	<колір> inherit	Колір тексту
opacity	<числове значення> inherit	Напівпрозорість елемента
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	Жирність тексту
font-style	normal italic oblique inherit	Стиль накреслення
text-decoration	none underline overline line-through blink inherit	Підчерки та інші лінії
font-variant	normal small-caps inherit	Вигляд великих літер
text-transform	capitalize uppercase lowercase none inherit	Зміна регістру тексту
line-height	normal <відстань> inherit	Висота стрічки
letter-spacing	normal <відстань>	Відстань між літерами
word-spacing	normal <відстань>	Відстань між словами
Розриви стрічок		
white-space	normal pre nowrap pre-wrap pre-line inherit	Задає правила виводу тексту та розривів
word-wrap	normal break-word inherit	Правила розриву слів
Вертикальне вирівнювання		
vertical-align	baseline sub super top text-top middle bottom text-bottom inherit	Вертикальне вирівнювання текст
Створення тіні у тексті		
text-shadow	none <колір> <горизонтальне зміщення> <вертикальне зміщення> [<радіус розмиття>]	Створення тіні
Параметри фону		
background-color	transparent <колір> inherit	Колір фону
background-image	none url("<адреса файлу зображення>")	Задає графічний фон
background-repeat	no-repeat repeat repeat-x repeat-y inherit	Замощення екрану
background-position	<горизонтальна позиція> [<вертикальна позиція>] inherit <i>Горизонтальна позиція:</i> <число> left center right <i>Вертикальна позиція:</i> <число> top bottom center	Позиція фонового зображення відносно елемента
background-attachment	scroll fixed	Рух фону відносно вмісту елемента

Параметри абзаців		
text-align	left right center justify inherit	Горизонтальне вирівнювання
text-indent	<відступ абзацу>	Задає «червону стрічку»
Параметри списків		
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	Вигляд маркера чи нумерації
list-style-image	none <адреса> inherit	Графічний маркер
list-style-position	inside outside inherit	Місцерозміщення маркера
Параметри відображення		
visibility	visible hidden collapse	Вказує, чи буде елемент відображатися
display	none block inline ... inherit	Вигляд елемента
Параметри курсора		
cursor	auto none default help pointer progress wait text	Вигляд курсора мишки
Параметри контейнерів		
width	auto <size> inherit	Ширина
height	auto <size> inherit	Висота
min-width	<size>	Мінімальна ширина
min-height	<size>	Мінімальна висота
max-width	<size>	Максимальна ширина
max-height	<size>	Максимальна висота
float	left right none inherit	Вирівнювання
clear	left right both none inherit	Розміщення знизу
overflow	visible hidden scroll auto inherit	Поведінка при переповненні
Параметри відступів елемента		
padding-left	auto <відступ> inherit	Внутрішній відступ зліва
padding-right	auto <відступ> inherit	Внутрішній відступ справа
padding-top	auto <відступ> inherit	Внутрішній відступ зверху
padding-bottom	auto <відступ> inherit	Внутрішній відступ знизу
padding	auto <відступ> inherit	Внутрішній відступ
margin-left	auto <відступ> inherit	Зовнішній відступ зліва
margin-right	auto <відступ> inherit	Зовнішній відступ справа
margin-top	auto <відступ> inherit	Зовнішній відступ зверху
margin-bottom	auto <відступ> inherit	Зовнішній відступ знизу
margin	auto <відступ> inherit	Зовнішній відступ
Параметри рамки елемента		
border-left-width	thin medium thick <товщина> inherit	Товщина рамки зліва
border-top-width	thin medium thick <товщина> inherit	Товщина рамки зверху
border-right-width	thin medium thick <товщина> inherit	Товщина рамки справа
border-bottom-width	thin medium thick <товщина> inherit	Товщина рамки знизу
border-width	thin medium thick <товщина> inherit	Товщина рамки
border-left-color	transparent <колір> inherit	Колір рамки зліва

border-top-color	transparent <колір> inherit	Колір рамки зверху
border-right-color	transparent <колір> inherit	Колір рамки справа
border-bottom-color	transparent <колір> inherit	Колір рамки знизу
border-color	transparent <колір> inherit	Колір рамки
border-left-style	none hidden dotted dashed solid double groove ridge inset outset inherit	Стиль рамки зліва
border-top-style	none hidden dotted dashed solid double groove ridge inset outset inherit	Стиль рамки зверху
border-right-style	none hidden dotted dashed solid double groove ridge inset outset inherit	Стиль рамки справа
border-bottom-style	none hidden dotted dashed solid double groove ridge inset outset inherit	Стиль рамки знизу
border-style	none hidden dotted dashed solid double groove ridge inset outset inherit	Стиль рамки
border-left	<товщина> <стиль> <колір> inherit	Задає всі параметри рамки зліва
border-top	<товщина> <стиль> <колір> inherit	Задає всі параметри рамки зверху
border-right	<товщина> <стиль> <колір> inherit	Задає всі параметри рамки справа
border-bottom	<товщина> <стиль> <колір> inherit	Задає всі параметри рамки знизу
border	<товщина> <стиль> <колір> inherit	Задає всі параметри рамки
Параметри виділення		
outline-width	thin medium thick <товщина> inherit	Товщина рамки виділення
outline-color	<колір> inherit	Колір рамки виділення
outline-style	none dotted dashed solid double groove ridge inset outset inherit	Стиль рамки виділення
outline	<товщина> <стиль> <колір> inherit	Параметри рамки виділення
Параметри таблиць		
text-align	left right center justify inherit	Горизонтальне вирівнювання в комірці
vertical-align	baseline sub super top text-top middle bottom text-bottom inherit	Вертикальне вирівнювання текст
padding-left	auto <відступ> inherit	Внутрішній відступ зліва
padding-right	auto <відступ> inherit	Внутрішній відступ справа
padding-top	auto <відступ> inherit	Внутрішній відступ зверху
padding-bottom	auto <відступ> inherit	Внутрішній відступ знизу
padding	auto <відступ> inherit	Внутрішній відступ
border-spacing	<відступ 1>[<відступ 2>]	Зовнішній відступ (якщо задано один відступ, то він зі всіх боків, якщо два – то перший зліва-справа, другий – зверху-знизу)
border-collapse	collapse separate inherit	Тип рамок комірок в таблиці
width	auto <size> inherit	Ширина комірки
height	auto <size> inherit	Висота комірки
table-layout	auto fixed inherit	Поведінка при переповненні комірки
caption-side	top bottom inherit	Місцерозміщення заголовка відносно таблиці
empty-cells	show hide inherit	Тип виведення порожніх комірок