

I. НАВЧАЛЬНІ ПИТАННЯ ТА РОЗРАХУНОК ЧАСУ

I. Вступна частина 5 хв

- викладач приймає доповідь чергового по групі;
- перевіряє наявність та зовнішній вигляд особового складу;
- підготовленість групи до занять та наявність навчальної літератури.

II. Основна частина 80 хв

1. Колір фона.
2. Фоновий малюнок.
3. Стилiзацiя гiперпосилань.
4. Стилiзацiя спискiв. Стилiзацiя форм.
5. Стилiзацiя табличних даних.

CSS представляє спеціальні властивості по стилізації списків. Одним з таких властивостей є **list-style-type**. Воно може набувати таких значень для нумерованих списків:

decimal: десяткові числа, відлік йде від 1;

decimal-leading-zero: десяткові числа, перед якими ставиться нуль, наприклад, 01, 02, 03, ... 98, 99;

lower-roman: рядкові латинські цифри, наприклад, i, ii, iii, iv, v;

upper-roman: заголовні латинські цифри, наприклад, I, II, III, IV, V;

lower-alpha: рядкові латинські літери, наприклад, a, b, c ...; **upper-alpha:** великі латинські літери, наприклад, A, B, C, ... Z;

Для ненумерованих списків:

disk: чорний диск;

circle: порожній кружечок;

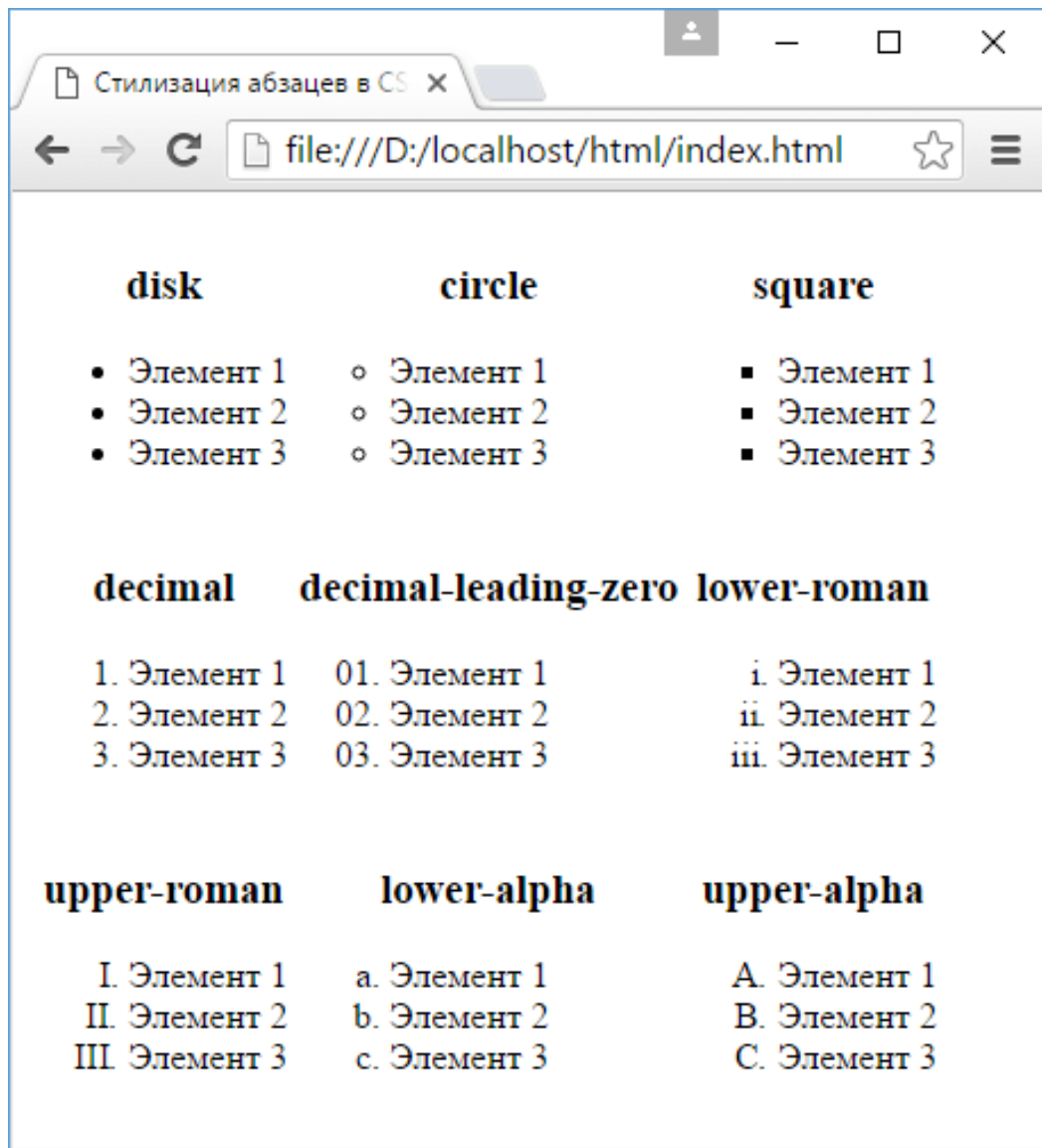
square: чорний квадратик.

Наприклад:

```
ul{  
    list-style-type: square;}
```

Щоб взагалі відключити маркери у елементів списку, використовується значення **none**:

```
ul{  
    list-style-type: none;}
```



Дана властивість може застосовуватися як до всього списку, так і до окремих елементів. Наприклад:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Стилізація списків в CSS</title>
    <style>
      .decimal { list-style-type: decimal; }
      ol { list-style-type: lower-roman; }
    </style>
  </head>
  <body>
    <ol>
      <li>Елемент 1</li>
      <li class="decimal">Елемент 2</li>
      <li>Елемент 3</li>
      <li>Елемент 4</li>
```

```
</ol>
</body>
</html>
```

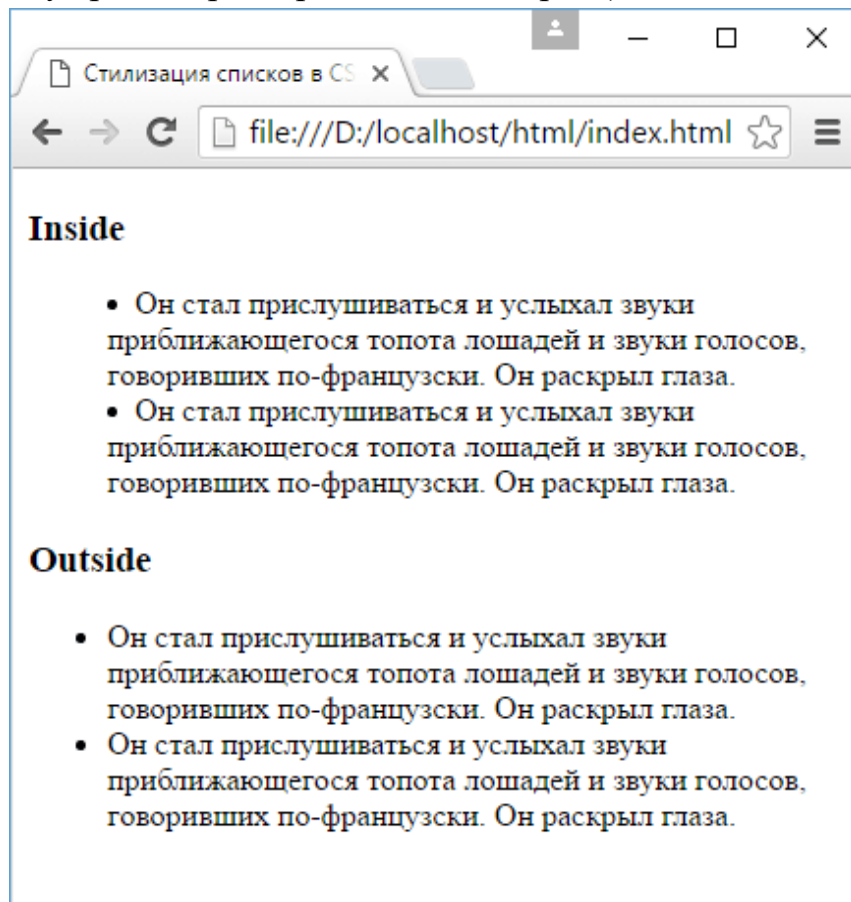
list-style-position

Веб-браузери зазвичай відображають маркери зліва від елементів списку. За допомогою властивості **list-style-position** ми можемо налаштувати їх позиціонування.

Дана властивість приймає два значення:

outside (за замовчуванням);

inside (забезпечує рівномірний розподіл по ширині).



```
<!DOCTYPE html>
```

```
<html>  <head>  <meta charset="utf-8">  <title>Стилізація списків в CSS</title>  <style>
```

```
ul.outside {list-style-position: outside;}
            ul.inside {list-style-position: inside; }
            </style>                                </head>
```

```
<body>
```

```
<h3>Inside</h3>
```

```
<ul class="inside">
```

```
<li>Він став прислухатися і почув звуки наближаємого тупоту коней та звуки голосів...</li>
```

```
<li> Він став прислухатися і почув звуки наближаємого тупоту коней та звуки голосів ...</li>
```

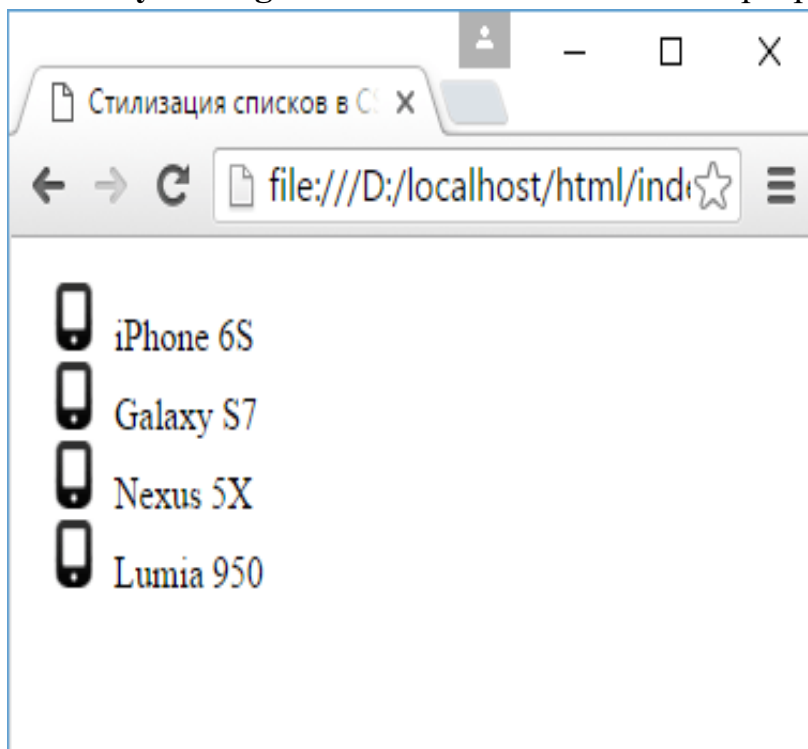
```

</ul>
<h3>Outside</h3>
<ul class="outside">
<li> Він став прислухатися і почув звуки наближаємого тупоту коней та звуки
голосів...</li>
<li> Він став прислухатися і почув звуки наближаємого тупоту коней та звуки
голосів...</li>
</ul>
</body>
</html>

```

list-style-image

Властивість **list-style-image** дозволяє задати в якості маркера зображення:



Властивість **list-style-image** як значення приймає шлях до зображення **url** (**phone_touch.png**), де "**phone_touch.png**" - це назва файлу зображення. Тобто в даному випадку мається на увазі, що в одній папці з веб-сторінкою знаходиться файл зображення **phone_touch.png**.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Стилізація списків в CSS3</title>
    <style>
      ul{
        list-style-image:url(phone_touch.png); }
    </style>
  </head>
  <body>

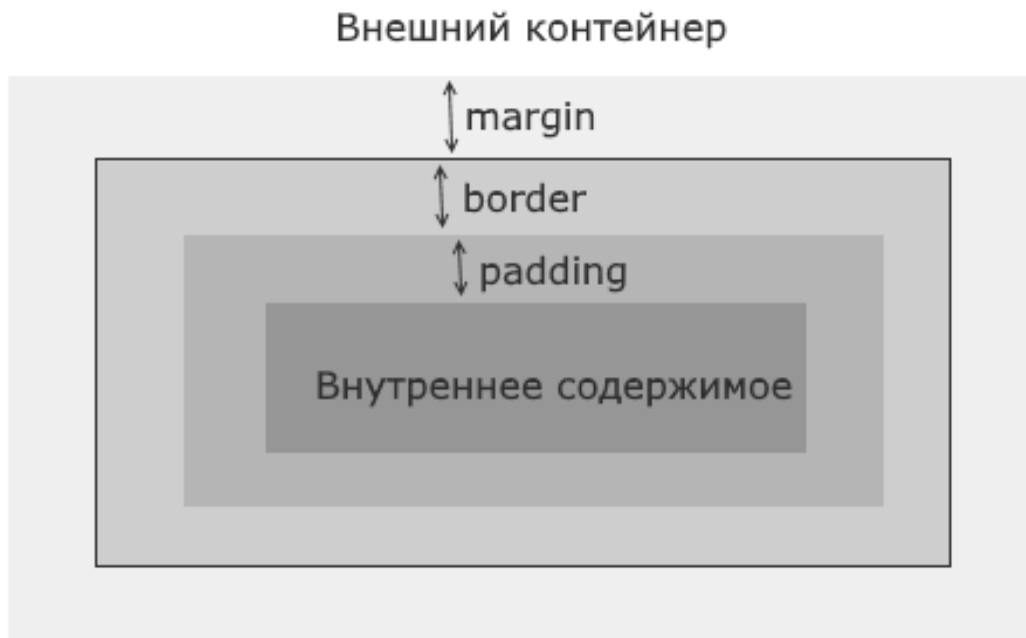
```

```
<ul>
  <li>iPhone 6S</li>
  <li>Galaxy S7</li>
  <li>Nexus 5X</li>
  <li>Lumia 950</li>
</ul>
</body>
</html>
```

Блочна модель

Для веб-браузера елементи сторінки представляють невеликі контейнери або блоки. Такі блоки можуть мати різний вміст - текст, зображення, списки, таблиці та інші елементи. Внутрішні елементи блоків самі виступають в якості блоків.

Схематично блокову модель можна представити таким чином:



Нехай елемент розташований в деякому зовнішньому контейнері. Це може бути елемент **body**, **div** та інш. Від інших елементів він відділяється певною відстанню - зовнішнім відступом, який описується властивістю CSS **margin**.

Тобто властивість **margin** визначає відстань від кордону поточного елемента до інших сусідніх елементів або до кордонів зовнішнього контейнера.

Далі починається сам елемент. І на початку йде його межа, яка в CSS описується властивістю **border**.

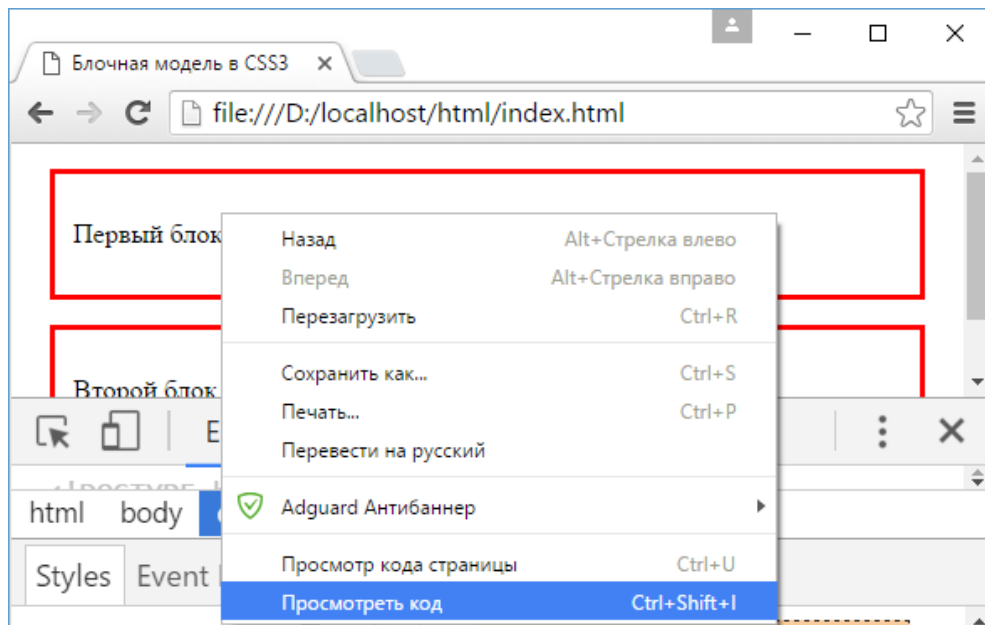
Після кордону йде внутрішній відступ, який в CSS описується властивістю **padding**. Внутрішній відступ визначає відстань від кордону елемента до внутрішнього вмісту.

Далі йде внутрішній вміст, який також реалізує ту ж блокову модель і також може складатися з інших елементів, які мають зовнішні і внутрішні відступи і кордон.

Наприклад: визначимо наступну веб-сторінку:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"> <title>Блочна модель в CSS</title>
    <style>
      div{
        margin: 15px; /* зовнішній відступ */
        padding: 11px; /* внутрішній відступ */
        border: 3px solid red; /* кордони шириною в 3 пікселя суцільної червоної
лінії */
      }
    </style> </head>
  <body>
    <div>
      <p>Перший блок</p>
    </div>
    <div>
      <p>Другий блок</p>
    </div>
  </body>
</html>
```

Після запуску веб-сторінки в браузері ми можемо подивитися блокову модель конкретних елементів. Для цього треба натиснути на потрібний елемент правою кнопкою миші і в контекстному меню вибрати пункт, який дозволяє переглянути вихідний код елемента. Для різних браузерів цей пункт може називатися по різному. Наприклад в Google Chrome це **Подивитися код**:



У даній моделі можемо побачити, як задаються відступи елемента, його межа, подивитися відступи від інших елементів і, за необхідністю, динамічно змінювати значення їх стилів.

Якщо явно не вказуємо значення властивостей **margin**, **padding** і **border**, то браузер застосовує встановлені значення.

Зовнішні відступи

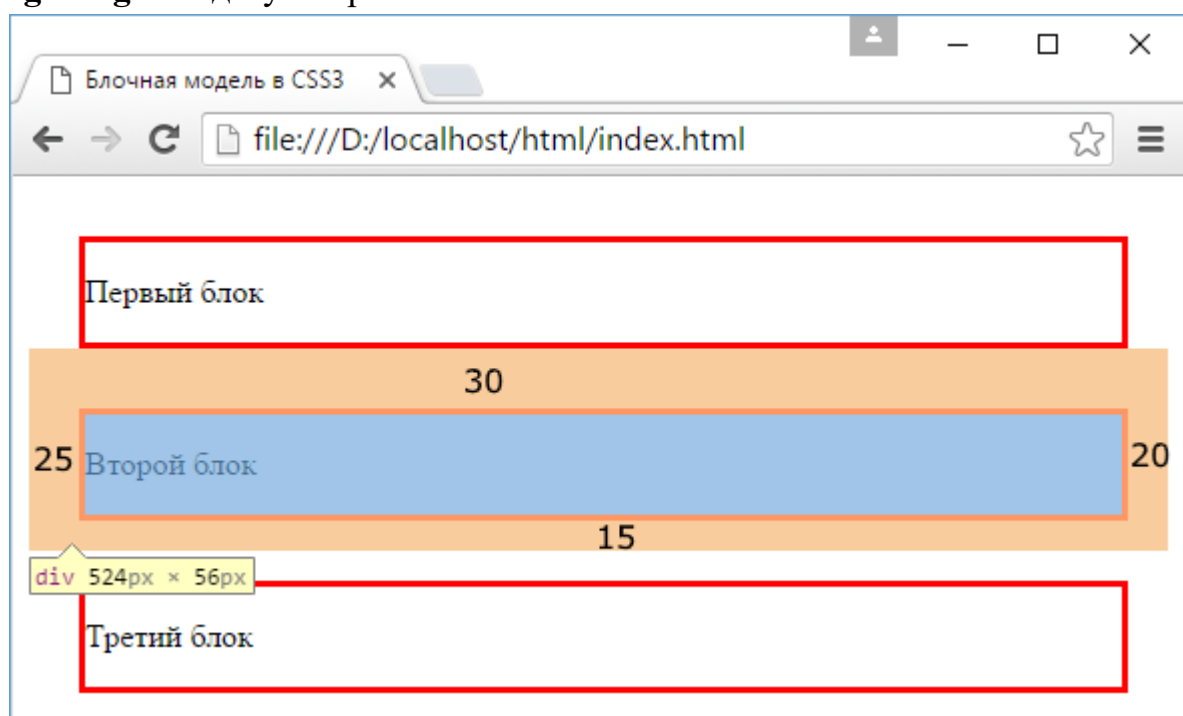
Властивість **margin** визначає відступ елемента від інших елементів або кордону зовнішнього контейнера. Існують спеціальні властивості CSS для завдання відступів для кожної сторони:

margin-top: відступ зверху

margin-bottom: відступ знизу

margin-left: відступ зліва

margin-right: відступ справа



Наприклад:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"> <title>Блочна модель в CSS</title>
    <style>
      div{margin-top: 30px; /* відступ зверху */
        margin-left: 25px; /* відступ зліва */
        margin-right: 20px; /* відступ зправа */
        margin-bottom: 15px; /* відступ знизу */
        border: 3px solid red; /* кордон */ }
    </style>
  </head>
  <body>
    <div> <p>Перший блок</p>    </div>
    <div> <p>Другий блок</p>    </div>
    <div> <p>Третій блок</p>    </div>
  </body>
</html>
```

Можна замість чотирьох властивостей задати одну:

```
div{
  margin: 30px 20px 15px 25px;
  border: 3px solid red; }
```

Властивість задається в форматі:

margin: відступ_зверху відступ_зправа відступ_знизу відступ_зліва;

Якщо значення для всіх чотирьох відступів збігається, то ми можемо вказати тільки одне значення:

```
div{
  margin: 25px; }
```

В цьому випадку для всіх чотирьох відступів буде використовуватися 25 пікселів. Для установки відступів можна використовувати точні значення в пікселях (px) або em, або відсоткове відношення, або значення auto.

Внутрішні відступи

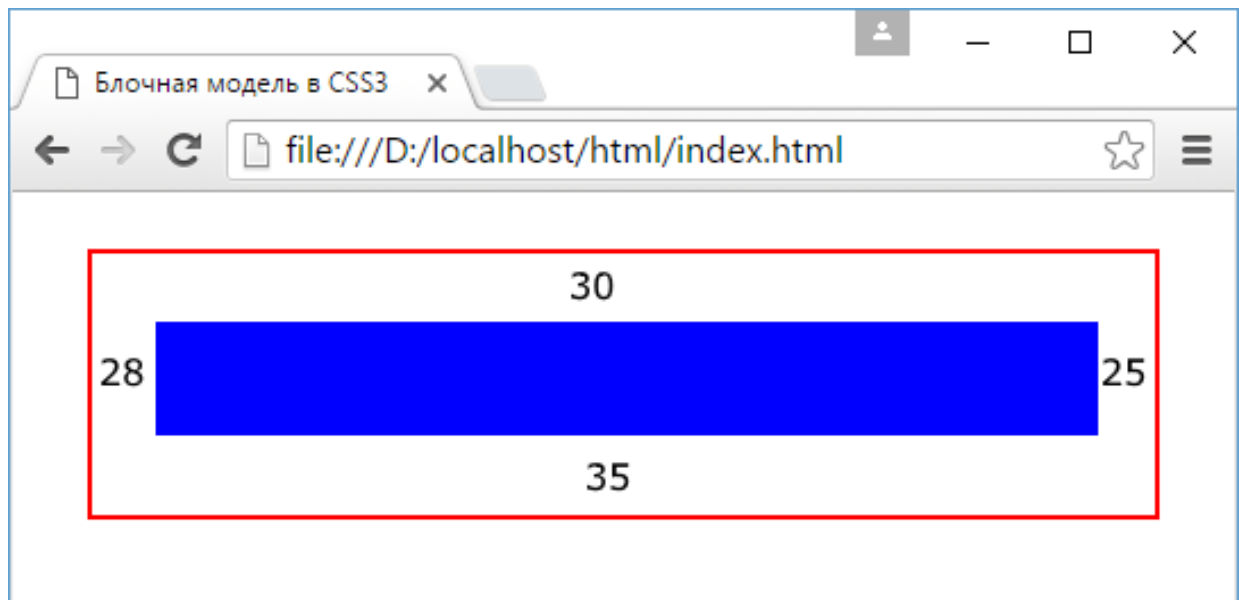
Властивість **padding** задає внутрішні відступи від кордону елемента до його внутрішнього вмісту. Як і для властивості **margin**, в CSS є чотири властивості, які встановлюють відступи для кожної зі сторін:

padding-top: відступ зверху

padding-bottom: відступ знизу

padding-left: відступ зліва

padding-right: відступ зправа



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"><title>Блочна модель в CSS</title>
    <style>
      div.outer{
        margin: 25px;
        padding-top:30px;
        padding-right: 25px;
        padding-bottom: 35px;
        padding-left: 28px;
        border: 2px solid red;    }
      div.inner{
        height: 50px;
        background-color:blue;    }
    </style>    </head>
  <body> <div class="outer">
    <div class="inner"></div>
  </div>
</body>
</html>
```

Для встановлення значення відступів, як і в **margin**, можуть застосовуватися або конкретні значення в пікселях, так і відсоткові значення (щодо розмірів елементів).

Для запису відступів також можна використовувати скорочений запис:

padding: відступ_зверху відступ_зправа відступ_знизу відступ_зліва;

```
div.outer{
  margin: 25px;
  padding: 30px 25px 35px 28px;
```

```
border: 2px solid red;  
}
```

Якщо всі чотири значення збігаються, то можна писати тільки одне значення для всіх відступів:

```
div.outer{  
margin: 25px;  
padding: 30px;  
border: 2px solid red;  
}
```

Кордони

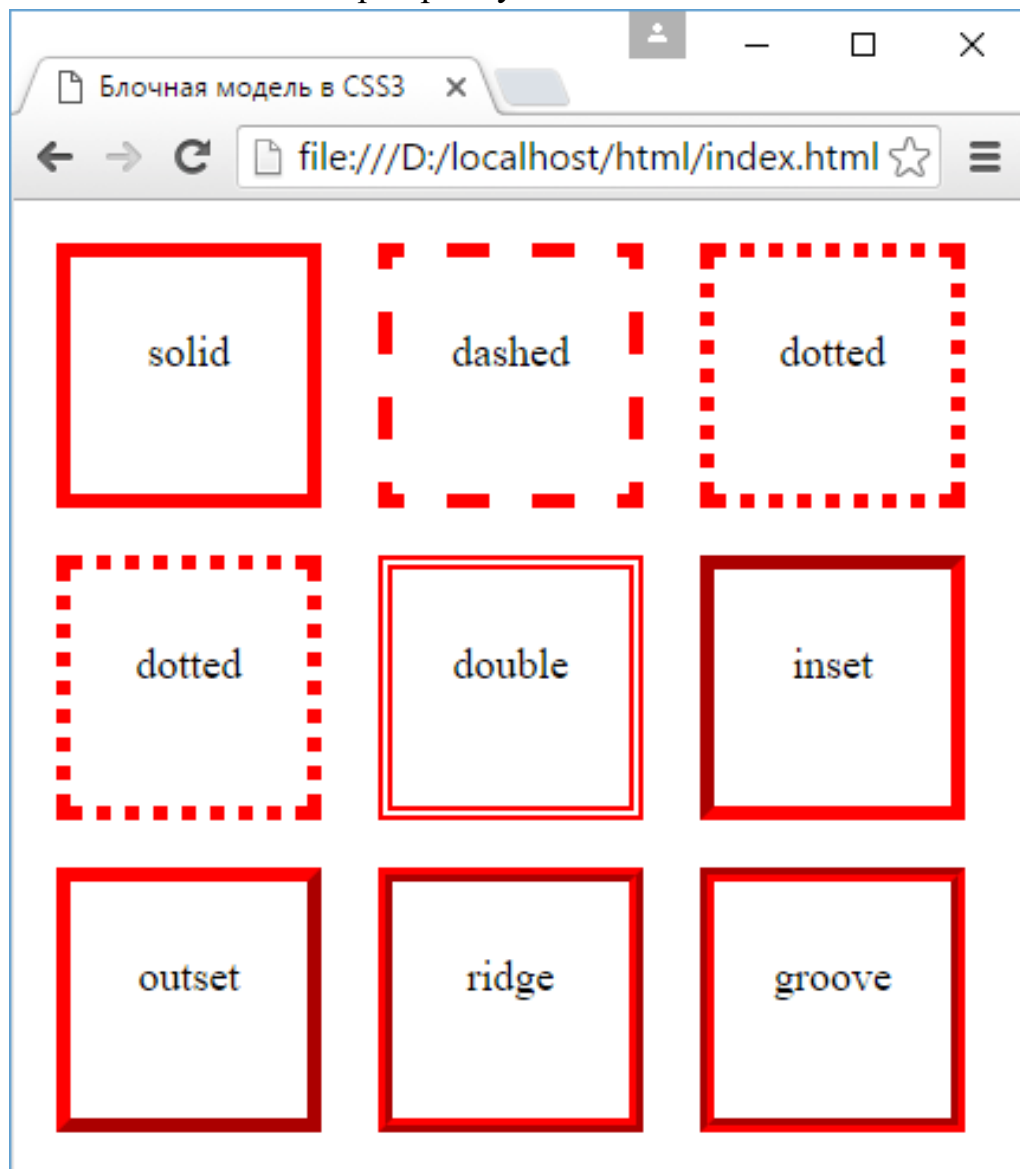
Кордон є елементом, який відокремлюється від зовнішнього по відношенню до нього вмісту. При цьому межа є частиною елемента.

Для налаштування кордону можуть використовуватися відразу кілька властивостей:

border-width: встановлює ширину кордону;

border-style: задає стиль лінії кордону;

border-color: встановлює колір кордону.



Властивість **border-width** може приймати наступні типи значень:

- Значення в одиницях вимірювання (**em**, **px** или **cm**):

Наприклад: **border-width: 2px;**

- Одне з константних значень: **thin** (тонкий кордон - 1px), **medium** (середня по ширині - 3px), **thick** (товста - 5px)

Наприклад: **border-width: medium;**

Властивість **border-color** в якості значення приймає колір CSS:

Наприклад: **border-color: red;**

Властивість **border-style** оформляє тип лінії кордону і може приймати одне з наступних значень:

none: межа відсутня

solid: межа у вигляді звичайної лінії

dashed: штрихова лінія

dotted: лінія у вигляді послідовності точок

double: межа у вигляді двох паралельних ліній

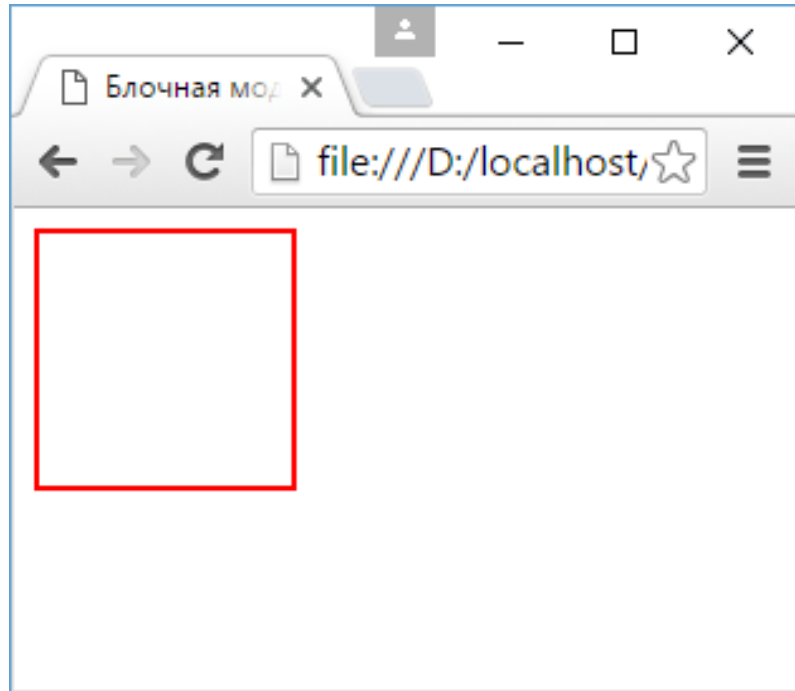
groove: межа має тривимірний ефект

inset: межа як би вдавлюється всередину

outset: аналогічно inset, тільки межа як би виступає назовні

ridge: межа також реалізує тривимірний ефект

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Блочна модель в CSS3</title>
    <style>
      div{
        width: 100px;
        height:100px;
        border-style: solid;
        border-color: red;
        border-width: 2px;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```



При необхідності ми можемо визначити колір, стиль і ширину кордону для кожної зі сторін використовуючи такі властивості:

/ для верхнього кордону */*

`border-top-width`

`border-top-style`

`border-top-color`

/ для нижнього кордону */*

`border-bottom-width`

`border-bottom-style`

`border-bottom-color`

/ для лівого кордону */*

`border-left-width`

`border-left-style`

`border-left-color`

/ для правого кордону */*

`border-right-width`

`border-right-style`

`border-right-color`

Властивість **border**

Замість встановлення окремо кольору, стилю і ширини кордону ми можемо використовувати одну властивість - **border:**

`border`: ширина, стиль, колір.

Наприклад:

`border: 2px solid red;`

Для встановлення кордону для окремих сторін можна використовувати одну з властивостей:

`border-top`

`border-bottom`

`border-left`

`border-right`

Їх використання за аналогією:

`border-top: 2px solid red;`

Радіус кордону

Властивість **`border-radius`** дозволяє округлити кордон. Ця властивість приймає значення радіуса в пікселях або одиницях **`em`**.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Блочна модель в CSS3</title>
```

```
    <style>
```

```
      div{
```

```
        width: 100px;
```

```
        height: 100px;
```

```
        border: 2px solid red;
```

```
        border-radius: 30px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

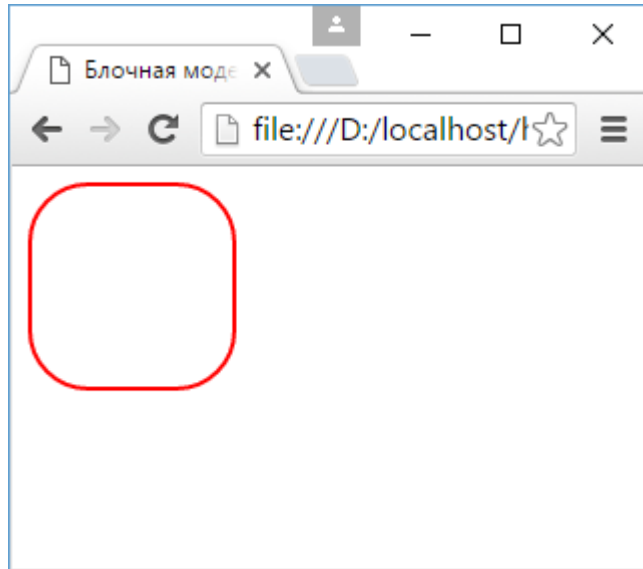
```
  <body>
```

```
    <div></div>
```

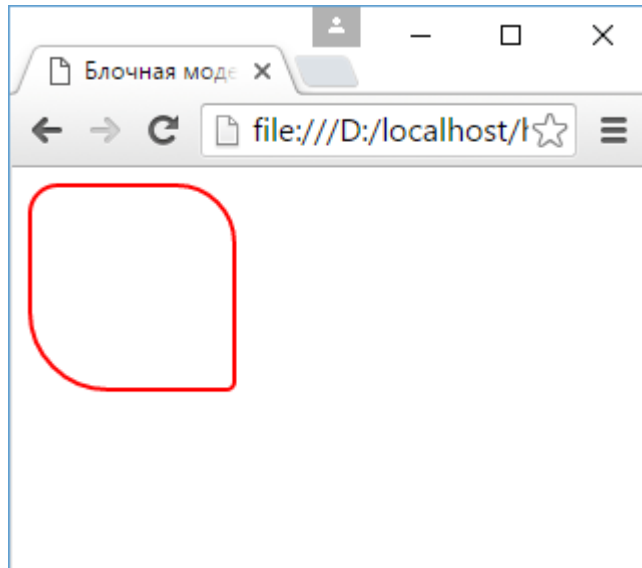
```
  </body>
```

```
</html>
```

Тепер кожен кут буде округлятися за радіусом в 30 пікселів:



Так як у елемента може бути максимально чотири кути, то можемо вказати чотири значення для встановлення радіусу кожного кута:
`border-radius: 15px 30px 5px 40px.`



Замість загальної установки радіусів для усіх кутів, можна їх встановлювати окремо.

Тобто попереднє значення **border-radius** можна переписати таким чином:
`border-top-left-radius: 15px; /* радіус для верхнього лівого кута */`
`border-top-right-radius: 30px; /* радіус для верхнього правого кута */`
`border-bottom-right-radius: 5px; /* радіус для нижнього лівого кута */`
`border-bottom-left-radius: 40px; /* радіус для нижнього правого кута */`

Також **border-radius** підтримує можливість створення еліптичних кутів. Тобто кут не просто округляється, а використовує два радіуси, утворюючи в результаті дугу еліпса:

`border-radius: 40px/20px;`

В даному випадку потрібно було, що радіус по осі X мав значення 40 пікселів, а по осі Y - 20 пікселів.