

Web-технології та web-дизайн. CSS

2024/2025

Лекція №4 Каскадні таблиці стилів (частина II)

Лекції: ст. викладач каф. Штучного інтелекту
Гриньова Олена Євгенівна
olena.hrynova@nure.ua

Зміст

CSS. Форматування тексту. Параметри тексту

Елементи `<div>` та ``

CSS Box Model (див. СР до Лк №4)

Рядкові та рядково-блокові елементи

CSS-анімація



CSS.

Форматування тексту

Параметри тексту

Параметри тексту: font-family

Для вибору виду шрифту в CSS-стилі використовується властивість **font-family** та встановлюється той шрифт, який ви хочете використати.

```
p {font-family: Arial;}
```

Властивість **font-family** працює тільки, якщо у відвідувачів сайтів є шрифт, встановлений на їх комп'ютерах!

Якщо у відвідувача вашого сайту не буде на його комп'ютері шрифту Arial, абзаци сторінки будуть відображені за допомогою початкового шрифту веб-браузера. Тому веб-дизайнери обмежуються невеликим набором шрифтів, які встановлені на більшості комп'ютерів заздалегідь. Оскільки ви не можете знати, чи є доступним вибраний вами шрифт у конкретного користувача, склалася практика вказувати не тільки основний шрифт, а й **пару резервних варіантів**.

```
p { font-family: Arial , Helvetica, sans-serif; "Times New Roman"; }
```

Якщо назва шрифту складається з декількох слів, вам необхідно взяти її в **лапки**.

Використання веб-шрифтів

Усі основні браузери підтримують використання **веб-шрифтів**. При цьому браузер фактично завантажує шрифт з веб-сервера та застосовує його для відображення тексту на веб-сторінці.

CSS-частина веб-шрифтів:

- правило **@font-face**, яке відповідає за повідомлення веб-браузеру як **назви шрифту**, так і **місця**, звідки його необхідно завантажити;
- властивість **font-family**, яка використовується з **веб-шрифтами** так само, як зі шрифтами, вже встановленими на комп'ютері.

```
@font-face {  
font-family: "League Gothic";  
src: url('fonts/League_Gothic-webfont.ttf');  
}
```

Параметри тексту: color

Параметр **color** можна встановити будь-яким наступним чином:

- Використання **ключового слова**. Ключові слова - це незалежні від регістру ідентифікатори, які представляють відповідний колір, наприклад **red**, **blue**, **black**, або **lightseagreen**. Ключове слово **transparent** - повністю прозорий колір (робить фон позаду кольорового елемента повністю видимим, тобто **transparent** - це `rgba(0,0,0,0)`). Ключове слово **currentColor** використовується в якості значення параметру **color** та приймає своє значення від успадкованого значення `color`-властивості.
- Використання **RGB cubic-coordinate системи**. Кольори RGB можуть бути виражені через шістнадцяткові (с префіксом **#**) та функціональні (`rgb()`, `rgba()`) нотації.
- Використання **HSL cylindrical-coordinate системи**. Модель кольорів HSL визначає заданий колір відповідно до його компонентів відтінка, насиченості і яскравості. Додатковий Альфа-компонент відповідає за прозорість кольору (`hsl()` і `hsla()`).

Параметри тексту: color

Встановлює колір тексту

Спадкування: так

Значенням **color** може бути:

- назва кольору

color: red;

- значення rgb

color: rgb(255,0,0);

color: rgb(100%,100%,100%);

color: rgba(255, 100, 50, .5);

- Шістнадцяткове значення

color: #3E8988;

- значення HSL

color: hsl(0, 100%, 50%);

color: hsla(0, 100%, 50%, .5);

<https://www.colourlovers.com/colors>

Параметри тексту: font-size

Властивість **font-size** встановлює розмір шрифту тексту. Після значення завжди має йти одиниця виміру величини. **Не треба додаватиме пробіл між значенням властивості та одиницею виміру.**

```
p {font-size: 36px;}
```

Для зміни розмірів різних елементів, в CSS використовуються **абсолютні** та **відносні** одиниці вимірювання.

Відносні одиниці, зазвичай, використовують для роботи с текстом, або коли необхідно обчислити відсоткове співвідношення між елементами:

em	Розмір шрифту поточного елемента
ex	Висота символу x
%	Відсоток

Параметри тексту: font-size

Абсолютні одиниці застосовуються не так часто, як відносні, та, зазвичай, використовують при роботі с текстом:

in	Дюйм (1 дюйм дорівнює 2,54 см)
cm	Сантиметр
mm	Міліметр
pt	Пункт (1 пункт дорівнює $1/72$ дюйма)
pc	Піка (1 піка дорівнює 12 пунктам)
px	1px дорівнює точно $1/96$ дюйма при будь-якому виведенні на друк незалежно від пристрою

Параметри тексту: font-size

Одиниці вимірювання не мають відношення до властивостей, але мають пряме відношення до **засобів** відображення: **екран** або **папір**.

	Рекомендовані	Іноді використовуються	Нерекомендовані
Таблиці стилів для виводу на екран	em, px, %	ex	pt, cm, mm, in, pc
Таблиці стилів для друку	em, cm, mm, in, pt, pc, %	px, ex	

Співвідношення між абсолютними одиницями таке:

$$1\text{in} = 2.54\text{cm} = 25.4\text{mm} = 72\text{pt} = 6\text{pc}$$

Так звані абсолютні одиниці (**cm**, **mm**, **in**, **pt** та **pc**) в CSS означають теж саме, що й усюди, але тільки якщо у пристрою виводу **достатньо висока роздільна здатність**.

Зараз абсолютні одиниці мають працювати правильно тільки при виводі на друк і на пристроях з **високою роздільною здатністю**.

Параметри тексту: font-size

Одиниці вимірювання: загальноприйняті **ключові слова** (**xx-small, x-small, small, medium, large, x-large** та **xx-large**), **em** (стандартна одиниця вимірювання в друкарській системі: розмір літери **m**, набраної шрифтом Cicero), **exs** (теж саме, але береться розмір літери **x**), **пікселі, відсотки, піки, пункти, дюйми, сантиметри й міліметри**.

Звичайний текст, який знаходиться поза межами тегів заголовків, відображається з висотою **16 пікселів** — це **основний (базовий) розмір шрифту** тексту.

Коли ви змінюєте текст за допомогою CSS, браузер бере базовий розмір шрифту тексту та корегує його в більшу або меншу сторону в залежності від значення ключового слова, одиниці вимірювання **em** (**1em** означає теж саме, що й **100%**), або відсотковим співвідношенням.

Параметри тексту: font-size

font-size: 36px;
font-size: large;
font-size: 200%;
font-size: 2.5em;

Будьте уважні, при використанні **em** та **відсоткових значень**, у випадку, **коли мають місце успадковані властивості розмірів шрифтів**. Якщо ви помітили, що якийсь текст виглядає не так, як передбачено, незвично великим або маленьким, переконайтесь в тому, що він не успадкував ці параметри.

У **CSS3** була додана нова одиниці вимірювання, що має назву **rem**. Ця назва означає Root EM, тобто її значення залежить від розміру тексту корінного (root) елемента.

font-size: .75**rem**;

Параметри тексту: font-style и font-weight

Курсив и **напівжирний** шрифт

Щоб виділити текст *курсивом*, додайте до стилю наступний код:

`font-style: italic;`

`font-style: oblique;`

Встановити для тексту звичайний, не курсивний шрифт:

`font-style: normal;`

Властивість товщини шрифту **font-weight** дозволяє робити текст **напівжирним** або звичайним. В CSS можна вказати дев'ять числових значень (100–900) для визначення точних, ледь помітних **градацій щільності** шрифту (от «супер щільної» (900) до «доволі легкої, майже невидимої» (100)).

`font-weight: 400;`

Щоб зробити шрифт тексту жирним, наберіть:

`font-weight: bold;`

Звичайний шрифт встановлюється наступним чином:

`font-weight: normal;`

Параметри тексту: text-transform (великі та малі літери)

Перетворення в прописні (ВЕЛИКІ) літери **uppercase**:

text-transform: uppercase;

Зробити всі літери малими можна за допомогою значення **lowercase**:

text-transform: lowercase;

Перетворити Перші Літери Слів На Великі — використовується **capitalize**:

text-transform: capitalize;

Оскільки ця властивість успадковується, будь-які теги, вкладені в **text-transform**, набувають того ж форматування: великі літери, малі, перші прописні літери слів. Щоб вказати CSS не змінювати регістр тексту, скористайтесь значенням властивості **none**:

text-transform: none;

Спадкування : так

Параметри тексту: прикрашання тексту

Властивість **text-decoration** для покращення зовнішнього вигляду тексту за допомогою додавання різних елементів. З її допомогою можна додати лінії підкреслення, надкреслення, перекрестити текст або зробити його мерехтливим. Властивість **text-decoration** застосовується разом з наступними ключовими словами: **underline**, **overline**, **line-through** або **blink**.

text-decoration: underline overline;

Використання мерехтіння **blink** (більша частина браузерів не зробить текст мерехтливим, навіть якщо ви вкажете їм це зробити)

text-decoration: blink;

Скасувати всі елементи прикрашання можна шляхом застосування ключового слова **none**:

text-decoration: none;

Спадкування: в Internet Explorer до версії 7.0 включно - ні

Параметри тексту: letter-spacing, word-spacing

Дозволяє збільшити або зменшити інтервал між символами

letter-spacing: -1px;

letter-spacing: .7em;

Звичайний інтервал між символами

letter-spacing: normal;

Інтервал між словами

word-spacing: 2px;

Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna
aliquam erat volutpat.

Спадкування: так

Можливе від'ємне значення

Параметри тексту: text-shadow

Властивість **text-shadow** дозволяє додавати до тексту тіні, для створення глибини й виразності заголовків, списків і абзаців. Властивість **text-shadow** потребує задання параметрів: **горизонтального зміщення** (де зліва чи справа від тексту має з'явитись тінь), **вертикального зміщення** (наскільки вище чи нижче тексту буде тінь), **ступінь розмитості тіні** та **колір тіні**.

Можна вказувати декілька параметрів тіні, поділяючи їх між собою **комою**. В CSS3 враховується наступний порядок: перша тінь у списку розміщується максимально зверху, остання в списку — в самому низу. В CSS2 порядок зворотній: перша тінь розміщується в самому низу, а остання зверху.

Параметри тексту: text-shadow

```
<head>
```

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

```
<title>text-shadow</title>
```

```
<style>
```

```
.shadowtext {
```

```
text-shadow: 1px 1px 2px black, 0 0 1em red; /* Параметри тіні */
```

```
color: white; /* Білий колір напису*/
```

```
font-size: 2em; /* Розмір напису*/
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="shadowtext">У хащах півдня жив би цитрус? Так, але фальшивий екземпляр!</p>
```

```
</body>
```

У хащах півдня жив би цитрус? Так, але фальшивий екземпляр!

Параметри тексту: line-height

CSS дозволяє встановлювати **міжрядковий інтервал** (інтерліньяж) — відстань, проміжок між базовими лініями двох сусідніх рядків тексту, — використовуючи властивість **line-height**.

line-height: 150%;

Стандартний міжрядковий інтервал - еквівалентний 120 %.

line-height: 1.5;

line-height: 1,5 em;

Спадкування: так

Успадковується не значення у відсотках, а **розраховане значення міжрядкового інтервалу.**

Параметри тексту: direction

direction - необхідний для сайтів, в яких має значення напрям тексту. Наприклад, при використанні арабського алфавіту читання відбувається справа наліво.

Властивість **direction** є достатньо універсальною і керує наступними можливостями:

- спосіб відображення тексту в блоці;
- порядок стовпців в таблиці;
- позиція смуги прокрутки в блоці;
- положення останнього висячого рядка текстового блока

Напрямок тексту зліва направо **ltr** (left to right)

`direction: ltr;`

Напрямок тексту справа наліво **rtl**

`direction: rtl;`

Спадкування: так

Параметри тексту: direction

Порядок стовпців в таблиці

```
<style>
  table {
    direction: rtl; /* Стовпці йдуть справа наліво */
  }
</style>
```

2003	2002	2001	2000	1999
57	74	62	34	16
47	56	72	69	4
86	34	79	73	7

Параметри тексту: text-align

Вирівнювання тексту в елементі **CSS2.1**

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

Вирівнює текст зліва:

`text-align: left;`

Вирівнює текст справа:

`text-align: right;`

Центрує текст:

`text-align: center;`

Розтягує на ширину блока (браузер в цьому випадку додає проміжки між словами):

`text-align: justify;`

Спадкування: так

Параметри тексту: text-align

Вирівнювання тексту в елементі CSS3

встановлює горизонтальне вирівнювання вмісту вбудованого рівня всередині елемента блоку або поля клітинки таблиці в горизонтальному напрямку

text-align: center | justify | left | right | start | end

Не змінює положення елемента:

text-align: auto;

Успадковує значення базового:

text-align: inherit;

Аналогічно значенню **left**, текст вирівнюється зліва (якщо **direction** було **ltr**):

text-align: start;

Аналогічно значенню **right**, текст вирівнюється зліва:

text-align: end;

Спадкування: так

Параметри тексту: text-indent

Абзацний відступ. Можна використовувати такі одиниці вимірювання, як **px** (пікселі), **em**, дюйми (**in**), пункти (**pt**) та інші.

Значення в пікселях — **абсолютне** значення, точніше число.

В той час як **em** визначає розмір відступу **в кількості символів** (базуючись на поточному розмірі шрифту).

`text-indent: 25px;`

`text-indent: 5em;`

Можливі від'ємні значення, для яких перший рядок буде зміщений вліво.

Спадкування: так

Параметри тексту: margin

Керування полями (межами) абзаців

Видалити проміжки або додаткові інтервали, які **будь-який браузер** додає **між абзацами**, можна за допомогою **margin-top** та **margin-bottom**.

Щоб повністю позбутись верхнього та нижнього полів, введіть наступне:

```
margin-top: 0;
```

```
margin-bottom: 0;
```

Щоб видалити поля між усіма абзацами веб-сторінки, створіть такий стиль:

```
p {margin-top: 0; margin-bottom: 0;}
```

Для встановлення значень абзацних полів, як і для відступів, ви можете застосовувати такі одиниці вимірювання, як **пікселі** або **em**.

Можна також використовувати відсотки, але **відсоткові значення** відносяться до **ширини елемента**, в якому знаходиться абзац.

Параметри тексту: форматування першої літери абзацу

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<style>
```

```
p:first-letter { font-size: 38px; font-weight: bold; color: red; }
```

```
</style>
```

```
</head>
```

```
<body>
```

`<p>` CSS дає можливість форматування абзацу з використанням псевдоелемента `:first-letter`

```
</p>
```

```
</body>
```

```
</html>
```

CSS дає можливість форматування абзацу з використанням псевдо елемента `:first-letter`

Параметри тексту: форматування першого рядка абзацу

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    p:first-line { font-weight: bold; color: red;}
  </style>
</head>
<body>
  <p> CSS дає можливість форматування першого рядка абзацу з використанням
    псевдоелемента :first-line
  </p>
</body>
</html>
```

CSS дає можливість форматування першого рядка абзацу з використанням псевдо елемента :first-line

CSS дає можливість форматування першого рядка абзацу з використанням псевдо елемента :first-line

Параметри тексту: white-space

Властивість **white-space** встановлює, як відобразити пробіли між словами. Будь-яка кількість пробілів в коді HTML показується на веб-сторінці як один. **white-space** імітує роботу тегу **<pre>**, але на відміну від нього не змінює шрифт на моноширокий.

Браузер ігнорує пробіли:

white-space: normal;

Текст відображається з урахуванням усіх пробілів і переносів, як вони були додані розробником в коді HTML. Якщо рядок виходить занадто довгим і не вміщається у вікні браузера, то буде додана **горизонтальна смуга прокрутки**:

white-space: pre;

Пробіли не враховуються, переноси рядків в коді HTML ігноруються, весь текст відображається одним рядком; разом з цим, додавання тегу **
** переміщує текст на новий рядок:

white-space: nowrap;

Спадкування: так

Параметри тексту : white-space

Фрагмент коду HTML

```
<head>
  <style>
    .example {
      white-space: pre;
      /* Враховуються всі пробіли та переноси*/
    }
  </style>
</head>
<body>
  <p class="example">
    &lt;html&gt;
    &lt;body&gt;
    &lt;b&gt; Велика теорема Ферма&lt;/b&gt;&lt;br&gt;
    &lt;i&gt;X &lt;sup&gt;&lt;small&gt;n&lt;/small&gt; &lt;/sup&gt;
    + Y &lt;sup&gt;&lt;small&gt;n&lt;/small&gt; &lt;/sup&gt; =
    Z &lt;sup&gt;&lt;small&gt;n&lt;/small&gt; &lt;/sup&gt; &lt;/i&gt;&lt;br&gt;
    де n - ціле число &gt; 2
    &lt;/body&gt;
    &lt;/html&gt;
  </p>
</body>
```

Як це виглядає у браузері

```
<html>
<body>
  <b> Велика теорема Ферма </b> <br>
  <i>X<sup><small>n</small> </sup>
  +Y<sup><small>n</small></sup>=
  Z<sup><small>n</small></sup></i><br>
  де n - ціле число > 2
</body>
</html>
```

Пояснення:

<; означає <

>; означає >

Елементи <div> та

Блокові та вбудовані елементи

Блоковим називається елемент, який відображається на веб-сторінці у вигляді прямокутника. Такий елемент **займає всю доступну ширину**, висота елемента залежить від його **вмісту**, і він **завжди починається з нового рядка**. До блокових елементів відносяться контейнери **<div>**, **<h1>**, **<p>** та інші.

Можна розміщувати один блоковий елемент всередині іншого, а також поміщати всередину них **вбудовані** елементи (****, наприклад).

Забороняється додавати всередину вбудованих елементів блокові.

Блокові та вбудовані елементи

За замовчуванням ширина блокового елемента розраховується автоматично и займає всю доступну ширину області, в якій розташовується.

Достатньо розмістити один тег **<div>** всередині іншого, як зразу ширина внутрішнього тегу починає розраховуватись відносно його **базового** елемента, тобто **зовнішнього** контейнера.

Робота с окремими блоками є набагато більш зручною, ніж при табличній верстці. До того ж при численних розміщеннях таблиць в інших таблицях досить легко помилитись.

Елементи `<div>` та ``

Для тегів `<div>` та `` нема ніяких властивостей для візуалізації.

Тег `<div>` (призначений для поділу на фрагменти) визначає будь-який окремий блок контенту, як, наприклад, абзац або заголовок. Проте ви також можете логічним чином об'єднати будь-який набір таких елементів.

При використанні CSS ви зможете **позиціонувати** будь-який блок в необхідному місці веб-сторінки, створюючи складну схему розмітки.

Тег `` застосовується до внутрішніх (**inline**) елементів сторінки, тобто до слів, фраз, які знаходяться **в межах абзацу** тексту або вмісту. Його можна використовувати так само, як інші внутрішні HTML-теги.

Блокові елементи: <div>

Елемент **<div>** - це **блоковий** елемент, який є необхідним для виділення фрагменту документа, з метою **змінити вигляд** контенту. Як правило, вид блока керується за допомогою стилів. Щоб не описувати кожен раз стиль всередині тегу, можна додати глобальний стиль, а для тегу додати атрибути **class** або **id**.

Так само, як і при використанні інших блокових елементів, **вміст тегу <div> завжди починається з нового рядка. Після нього також додається перенос рядка.**

<div>...</div>

Атрибути

align задає вирівнювання змісту тегу <div>.

title додає спливаючу підказку до змісту тегу <div>.

Також доступні **універсальні** атрибути и події.

Багато нових тегів HTML5 призначені для розширення можливостей звичайного тегу <div>.

Рядкові елементи: ``

Тег `` призначений для визначення **рядкових** елементів документа.

`...`

Атрибути

Для цього тегу доступні **універсальні** атрибути и події.

Щоб не описувати кожен раз стиль всередині тегу ``, можна виділити стиль в глобальну таблицю стилів, а для тегу додати атрибут **class** або **id**.

Рядковий елемент `` може вміщати в собі **тільки дані або інші рядкові елементи** (**рядкові елементи ніяк не можуть зберігати блокові елементи**).

Ширина рядкового елемента `` дорівнює його вмісту плюс значення відступів, полів та меж.

Типи елементів

- Блокові
- Рядкові (лінійні)
- Рядково-блокові

Блокові елементи

Блокові елементи – елементи, контент яких відображається на сторінці у вигляді прямокутних блоків. Блокові бокси за замовчуванням мають такі теги як:

<details>, <blockquote>, <div>, , <p>, <table>, , <dl>, , <h1>, <h2>, <hr> та інші.

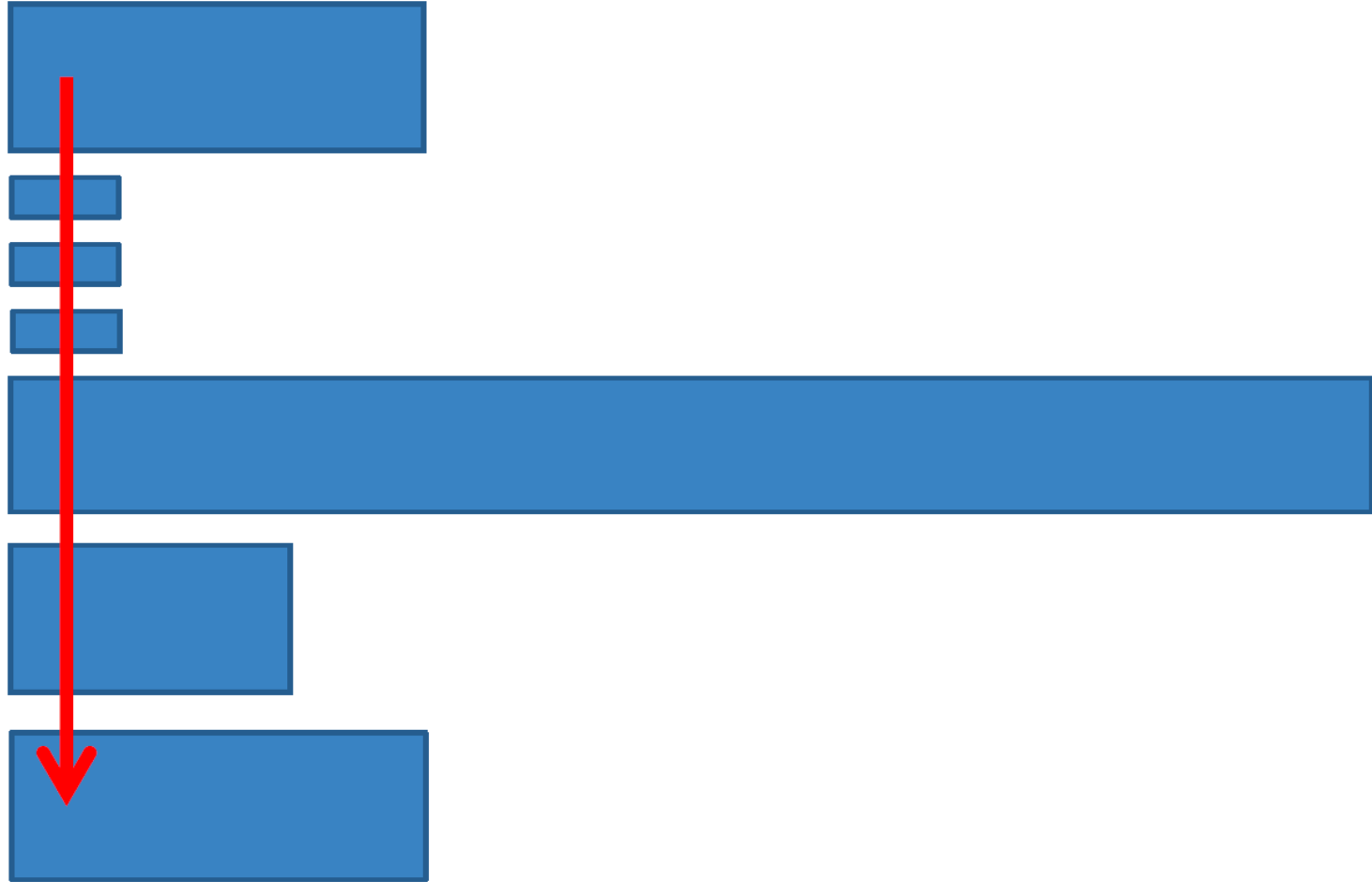
Поведінка блокових елементів

До и після блокового елемента існує перенос рядка.

Блоковим елементам можна задати ширину, висоту, внутрішні та зовнішні відступи.

Займають увесь доступний простір по горизонталі в контейнері.

Поведінка блокових елементів



Поведінка рядкових елементів

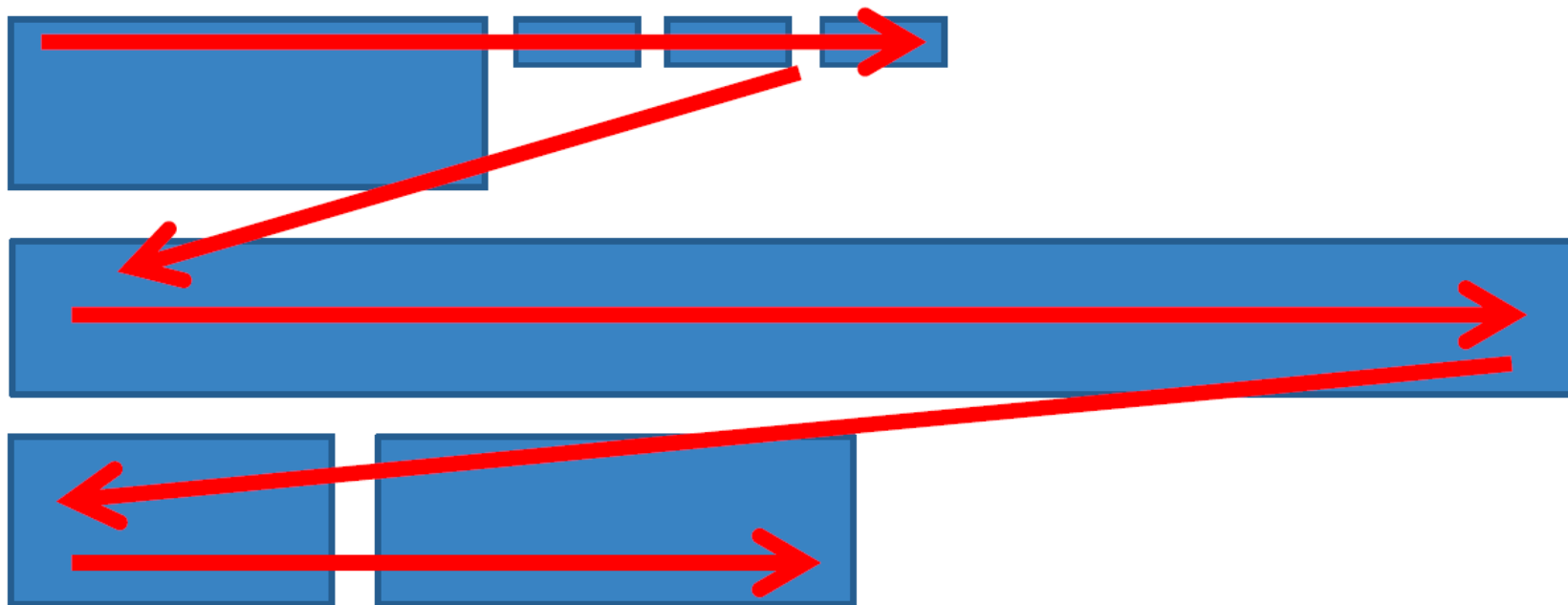
Рядкові бокси — протилежність блоковим. Рядкові бокси розташовуються один за одним в одному рядку, можуть розриватись і знаходитись на декількох рядках.

До і після рядкового елемента відсутні переноси рядків.

Ширина та висота рядкового елемента залежить тільки від його змісту, задати розміри за допомогою CSS неможливо. Можна задавати **тільки горизонтальні відступи**.

Наприклад: `<a>`, ``, ``, ``, ``

Поведінка рядкових елементів



CSS-анімація

CSS- анімація

Деякі властивості CSS можна анімувати , тобто їх можна використовувати в анімаціях і переходах.

Анімовані властивості можуть поступово змінюватися від одного значення до іншого, наприклад розмір, числа, відсоток і колір.

https://www.w3schools.com/cssref/css_animatable.asp

CSS

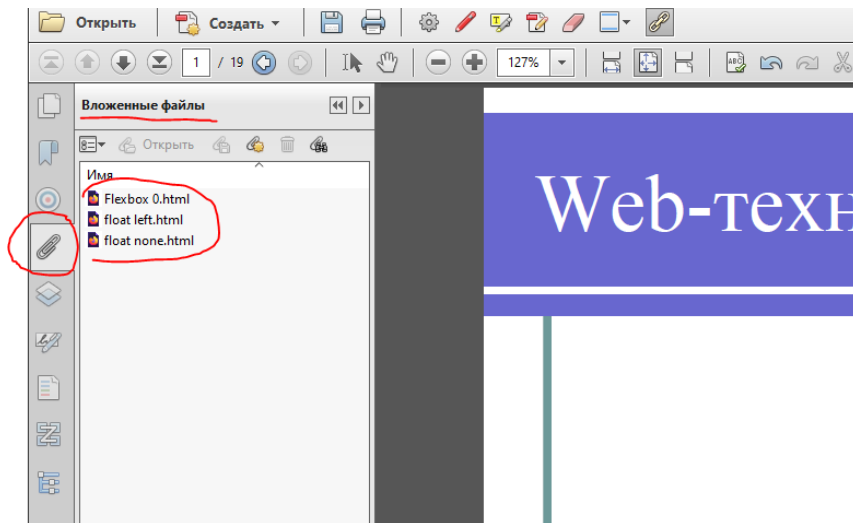
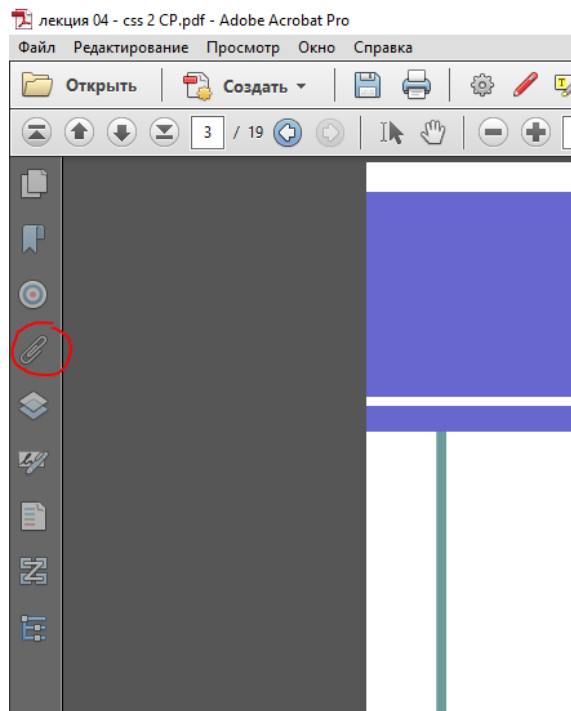
```
#myDIV { width: 300px;
height: 200px;
background: red;
animation: mymove 5s infinite;
@keyframes mymove { from {background-color: red;}
to {background-color: blue;} }
```

HTML

```
<h1>Animation of background-color</h1>
<p>Gradually change the background-color from red to blue:<p>
<div id="myDIV"></div>
<p><strong>Note:</strong> CSS Animations do not work in Internet Explorer 9 and
earlier versions.</p>
```

CSS- анімація

Перегляньте файли у вкладенні цього файлу.



Дякую за увагу!