

Основи CSS

Завдання теоретичне

Вивчити теорію CSS (підключення, селектори, властивості та значення).
Більш детально про CSS у кінці цього файлу.

Завдання (практичне) лабораторної роботи

1. До html сторінки, створеної в лаб.1, підключити **один** зовнішній файл css. Для цього створити файл style.css і в html файлі прописати link на цей файл.
2. Блокам з першої лабораторної дописати класи та ідентифікатори.
3. В css файлі використати не менше 7 різноманітних видів селекторів. Наприклад, селектор по класу, по id, груповий...
4. В css файлі блокам з відповідними селекторами дописати різноманітні властивості.
5. Використати не менше 20 різних властивостей. Наприклад: margin, padding...
6. Застосувати властивості: для тексту, box-моделі, гіперпосилань, списків, display, overflow, inline-block, pseudo-class, opacity та інші.
7. Вивчити яких значень можуть набувати різноманітні властивості.
8. Написати css код використовуючи внутрішнє та вбудоване розміщення css коду.

Теорія

Список селекторів та властивостей

Теорія

Основи CSS

CSS (Cascading Style Sheets), або каскадні таблиці стилів, описують правила форматування окремого елемента веб-сторінки. Створивши стиль один раз, його можна застосовувати до будь елементів сторінки скільки завгодно разів.

Визначення стилю складається з двох основних частин: самого елемента веб-сторінки - **селектора**, і команди форматування - **блоку оголошення**.

Селектор повідомляє браузеру, який саме елемент форматувати, в блоці оголошення перелічено властивості команд форматування і їх значення.



Рис. 1. Структура оголошення стилю елемента в CSS

Додавання CSS до веб-сторінки

1.1. Вбудовані таблиці стилів

Вбудовані стилі представляють набір стилів, що є частиною коду веб-сторінки, знаходяться між тегами `<style> ... </ style>` і розташовані всередині елемента `<head>`. Вбудовані стилі діють тільки на сторінці, на якій вони містяться. Вбудовані стилі мають пріоритет перед глобальними, але поступаються стилям, що вказані в тезі. На одній сторінці можна розміщувати довільну кількість вбудованих стилів:

```
<head>
<style type = "text / css">
h1, h2 { color: red; font-family: "Times New Roman", Georgia, Serif; line-height: 1.3em;}
</ style>
</ head>
```

1.2. Внутрішні текстові таблиці стилів

Внутрішні текстові стилі не використовують селектори, присвоєння стилю відбувається безпосередньо до html-елементу через атрибут `style`:

```
<p style = "font-family:" Times New Roman ", Georgia, Serif; color: # 70d7700;"> Зверніть увагу на цей текст. </ P>
```

Недоліком такого способу є неможливість автоматичного використання даного стилю для іншого елемента.

1.3. Правило @import

Правило `@import` дозволяє завантажити зовнішню таблицю стилів. Щоб директива `@import` працювала, вона повинна розташовуватися всередині тега `<style>` перед іншими правилами:

```
<style type = "text / css">
@import url (mobile.css);
p { font-size: 0.9em; color: grey;}
</ style>
```

1.4. Тег <link>

Зовнішня таблиця стилів представляє текстовий файл з розширенням .css, в якому знаходиться весь набір стилів CSS. Він не містить HTML-код, тому його не потрібно вкладати всередину тегів <style> ... </style>. Задані в файлі стилі будуть працювати для всіх сторінок веб-сайту.

Зовнішня таблиця стилів (кілька таблиць стилів) прикріплюється до веб-сторінки за допомогою тега <link>, вложеного в тег <head>:

```
<head>
<link rel = "stylesheet" type = "text / css" href = "style/style1.css">
<link rel = "stylesheet" type = "text / css" href = "style/style2.css">
</ head>
```

де rel = "stylesheet" вказує тип посилання (посилання на таблицю стилів), а type = "text / css" повідомляє браузеру тип даних, в даному випадку це текстовий файл, що містить css-код.

2. Селектори

За допомогою селекторів створюються CSS-правила для форматування тегів (елементів) веб-сторінки. В якості селекторів використовуються не тільки самі теги (елементи) та їхні класи і ідентифікатори, а також псевдокласи і псевдоелементи.

Псевдокласи дозволяють додавати особливі класи до елементів, вибираючи об'єкти, яких немає в структурі веб-сторінки, або які не можна вибрати за допомогою звичайних селекторів, наприклад, перша буква або перший рядок одного абзацу.

Псевдоелементи вибирають елементи, які не є елементами структури html-сторінки.

Список селекторів CSS

Універсальний селектор. Універсальний селектор відповідає будь-якому тегу (елементу), наприклад, наступний запис обнулить відступи для всіх елементів веб-сайту:

```
* {Margin: 0;}
```

Селектор Тегу(Елемента). Селектори тегів (елементів) використовуються для визначення стилів елементів для всіх сторінок веб-сайту, наприклад, стиль заголовків h1 або загальний стиль абзаців:

```
h1 {font-family: Lobster, cursive;}
```

```
p {letter-spacing: 0.1em;}
```

Селектор Класу. Селектори класу використовуються для визначення стилів для кількох тегів (елементів) одного типу, розміщених у різних частинах або на різних сторінках веб-сайту. Для створення заголовка класу headline необхідно додати атрибут class з відповідним значенням в відкриваючий тег <h1>. Далі необхідно задати стиль для зазначеного класу. Даний стиль оформлення можна застосувати і для інших елементів.

Код HTML	Код CSS
<h1 class = "headline"> Інструкція користування персональним комп'ютером </ h1>	headline { text-transform: uppercase; color: lightblue; }

Селектор Ідентифікатора. Селектори ідентифікатора використовуються для привласнення стилю одному конкретному тегу (елементу). Ідентифікатор id елемента можна використовувати в документі лише один раз, оскільки він виділяє унікальний елемент.

```
#sidebar { text-transform: uppercase; color: lightblue; }
```

Селектор Нащадка. До нащадків елемента відносяться його дочірні елементи. Селектори нащадків дозволяють стилізувати всі вкладені елементи, наприклад, можна відформатувати зовнішній вигляд елементів маркованого списку:

```
ul li {text-transform: uppercase;} - вибере всі елементи li, що є нащадками всіх елементів ul.
```

Якщо потрібно відформатувати нащадки певного елемента, то можна поставити йому стильовий клас:

p.first a {color: green;} - означає, що потрібно застосувати даний стиль до всіх посилань, нащадків абзацу, що відноситься до класу з ім'ям first;

p .first a {color: green;} - якщо додати пробіл, то будуть обрані посилання, розташовані всередині будь-якого тегу класу .first, який є нащадком елемента <p>;

.first a {color: green;} - даний стиль застосовується до любого посилання, що розташоване всередині інших тегів, позначених класом .first.

Дочірній селектор. Дочірній тег є прямим нащадком тегу, що його містить. Тобто, відносини "батьки-діти" існують між елементами і тими елементами, які містяться безпосередньо всередині них. В одного елемента може бути кілька дочірніх елементів, а батьківський елемент може бути в кожного елемента тільки один.

p> strong - вибирає всі елементи strong, які є дочірніми по відношенню до елементу p.

Сестринський селектор. Сестринські відносини виникають між елементами, що мають загального батька. Селектори сестринських елементів дозволяють вибрати теги з групи елементів одного рівня.

h1 + p - вибере всі перші абзаци, що йдуть безпосередньо за будь-яким тегом <h1>, не зачіпаючи решта абзаців.

h1 ~ p - вибере всі абзаци, які є сестринськими по відношенню до будь-якого заголовку h1 і йдуть після нього.

Селектор Атрибута. Селектори атрибутів дозволяють формувати елементи на основі вибірки будь-яких містяться в них атрибутів або значень атрибутів, варіанти:

[атрибут] - вибирає всі елементи, для яких заданий вказаний атрибут.

img [alt] - вибирає всі картини, що містять атрибут alt.

img [title = "flower"] - вибирає всі картини, назва яких містить слово flower.

Псевдоклас. Псевдокласи - це класи, що фактично не прикріплені до тегів html-коду. Вони викликають CSS-правила при вчиненні тої чи іншої події або підкоряються певному правилу:

a: link - посилається на невідвідане посилання.

a: visited - посилається на вже відвідане посилання.

a: hover - посилається на будь-який елемент, над яким проводять курсором миші.

a: focus - посилається на будь-який елемент, над яким знаходиться курсор миші.

a: active - посилається на елемент, який був активізований користувачем.

Структурні псевдокласи. Структурні псевдокласи форматують дочірні елементи відповідно до зазначеного параметра в скобці, варіанти:

:nth-child (odd) - вибирає непарні дочірні елементи.

:nth-child (even) - вибирає парні дочірні елементи.

:nth-child (3n) - вибирає кожен третій елемент серед дочірніх.

Структурні псевдокласи типу. Дозволяють вказати на конкретний тип дочірнього тегу:

:nth-of-type () - вибирає елементи за аналогією з: nth-child (), при цьому бере до уваги тільки тип елемента.

:first-of-type - дозволяє вибрати перший дочірній елемент.

:last-of-type - вибирає останній тег конкретного типу.

:nth-last-of-type () - вибирає елемент заданого типу в списку елементів відповідно до зазначеного місцеположенням, починаючи з кінця.

:only-of-type - вибирає єдиний елемент зазначеного типу серед дочірніх елементів батьківського елемента.

Псевдоелементи. Псевдоелементи використовуються для додавання вмісту, який генерується за допомогою властивості content, і для зміни зовнішнього вигляду частині елемента:

:before - вставляє згенерований вміст перед елементом.

:after - додає згенерований вміст після елемента.

2.1. Комбінація селекторів

Щоб домогтися більш чіткого вибору елементів для форматування, можна не обмежуватися завданням одного типу селектора, а використовувати комбінації селекторів, наприклад:

a [href] [title] - вибере всі посилання, для яких задані атрибути href і title;

img [alt * = css]: nth-of-type (even) - вибере всі парні картинки, альтернативний текст яких містить слово css.

2.2. Угрупування селекторів

Можна застосувати один і той же стиль до кількох елементів, причому обмежень за кількістю доданих елементів немає. Для цього необхідно в лівій частині оголошення помістити через кому потрібні селектори, наприклад:

```
h1, h2, h3, h4 {color: tomato; background: white;}
```

Ще більше селекторів

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

3. Принцип каскадування і специфічність правила

Каскадування представляє процес застосування різних правил до одного і того ж елементу. Більш конкретні правила мають пріоритет над більш загальними. Якщо у відношення одного і того ж елемента визначено кілька стилів, то в результаті до нього буде застосований останній з них.

Для кожного правила браузер обчислює специфічність селектора, і якщо у елемента є конфліктуючі оголошення властивостей, до уваги береться правило, що має найбільшу специфічність.

Значення специфічності складається з чотирьох частин: 0, 0, 0, 0. Специфічність селектора визначається наступним чином:

для id додається 0, 1, 0, 0;

для class додається 0, 0, 1, 0;

для кожного елементу і псевдоелемента додається 0, 0, 0, 1;

для вбудованого стилю, доданого безпосередньо до елементу - 1, 0, 0, 0;

універсальний селектор не має специфічності.

CSS

```
h1 {color: lightblue;} / * специфічність 0, 0, 0, 1 * /
```

```
em {color: silver;} / * специфічність 0, 0, 0, 1 * /
```

```
h1 em {color: gold;} / * специфічність: 0, 0, 0, 1 + 0, 0, 0, 1 = 0, 0, 0, 2 * /
```

```
div # main p.about {color: blue;} / * специфічність: 0, 0, 0, 1 + 0, 1, 0, 0 + 0, 0, 0, 1 + 0, 0, 1, 0 = 0, 1, 1, 2 * /
```

```
.sidebar {color: grey;} / * специфічність 0, 0, 1, 0 * /
```

```
#sidebar {color: orange;} / * специфічність 0, 1, 0, 0 * /
```

```
li # sidebar {color: aqua;} / * специфічність: 0, 0, 0, 1 + 0, 1, 0, 0 = 0, 1, 0, 1 * /
```

В результаті до елементу застосуються ті правила, специфічність яких більше, наприклад, якщо на елемент діють дві специфічності зі значеннями 0, 0, 0, 2 і 0, 1, 0, 1, то виграє друге правило.

Вагу правила також можна задати за допомогою ключового слова **!important**, яке додається після значення властивості, наприклад, **font-weight: bold!important**;

Правило необхідно розміщувати в кінець оголошення перед закриваючою дужкою. Таке оголошення буде мати пріоритет над усіма іншими правилами.

Властивості CSS2 за алфавітом

1. background
2. background-attachment
3. background-color
4. background-image
5. background-position
6. background-repeat
7. border
8. border-bottom
9. border-bottom-color
10. border-bottom-style
11. border-bottom-width
12. border-collapse

13. border-color
14. border-left
15. border-left-color
16. border-left-style
17. border-left-width
18. border-right
19. border-right-color
20. border-right-style
21. border-right-width
22. border-spacing
23. border-style
24. border-top
25. border-top-color
26. border-top-style
27. border-top-width
28. border-width
29. bottom
30. caption-side
31. clear
32. clip
33. color
34. content
35. counter-increment
36. counter-reset
37. cursor
38. direction
39. display
40. empty-cells
41. float
42. font
43. font-family
44. font-size
45. font-style
46. font-variant
47. font-weight
48. height
49. left
50. letter-spacing
51. line-height
52. list-style
53. list-style-image
54. list-style-position
55. list-style-type
56. margin
57. margin-bottom
58. margin-left
59. margin-right
60. margin-top
61. max-height
62. max-width
63. min-height
64. min-width
65. orphans
66. outline

- 67. outline-color
- 68. outline-style
- 69. outline-width
- 70. overflow
- 71. padding
- 72. padding-bottom
- 73. padding-left
- 74. padding-right
- 75. padding-top
- 76. page-break-after
- 77. page-break-before
- 78. page-break-inside
- 79. position
- 80. quotes
- 81. right
- 82. table-layout
- 83. text-align
- 84. text-decoration
- 85. text-indent
- 86. text-transform
- 87. top
- 88. vertical-align
- 89. visibility
- 90. white-space
- 91. widows
- 92. width
- 93. word-spacing
- 94. z-index

Нижче наведені часто вживані властивості та значення CSS.

Колір і фон (тло)

background - параметр дозволяє встановити одночасно до п'яти атрибутів стилю фону. Значення можуть іти в будь-якому порядку, браузер сам визначить, яке з них відповідає потрібного атрибуту. Простішими словами задає колір для нашого сайту.

background-attachment - параметр background-attachment встановлює, чи буде прокручуватися фонове зображення разом із вмістом сайт. Зображення може бути зафіксовано і залишатися нерухомим, або переміщатися разом з документом.

background-color - встановлює колір тла для сайту.

background-image - встановлює зображення для тла (фону сайту).

Наведу приклад:

```
BODY {  
  background-image: url(images/bg.jpg); /*вказуємо шлях до нашого зображення*/  
  background-color: #c7b29a; /*колір тла*/  
}
```


background-position - встановлює початкове положення фонового зображення, встановленого за допомогою параметра background-image.

background-repeat - визначає, як буде повторюватися фонове зображення, встановлене за допомогою параметра background-image, і по якій осі. Можна встановити повторення малюнка тільки по горизонталі, по вертикалі або в обидві сторони.

color - задає колір тексту.

opacity - визначає рівень прозорості для елемента. opacity 0.5; 0.1; це вже як потрібно налаштувати підбираєм під свій смак.

Межі

border - параметр одночасно дозволяє встановити властивості для товщини, стилю і кольору рамки навколо елемента.

border-bottom - параметр одночасно дозволяє встановити властивості: товщину, стиль і колір рамки внизу елемента.

border-bottom-color - встановлює колір внизу елемента.

border-bottom-style - встановлює стиль нижньої границі елемента.

border-bottom-width - встановлює товщину нижньої границі елемента.

border-color - задає колір на всіх границях елемента. Також можна задати тільки на окремих частинах.

border-left - параметр дозволяє одночасно встановити товщину, стиль і колір лівої рамки елемента. Значення можуть йти в будь-якому порядку, розділяючись пробілом, браузер сам визначить, яке з них відповідає потрібному атрибуту.

border-left-style - встановлює стиль межі ліворуч від елемента.

border-left-color - встановлює колір границі зліва від елемента.

border-left-width - задає товщину межі зліва від елемента.

border-top, border-top-color, border-top-style, border-top-width, border-right, border-right-color, border-right-style, border-right-width - вони аналогічні з border-left і border-bottom. left - з ліва, right - з права, top - з верху, bottom - знизу.

border-style - встановлює стиль рамки навколо елемента.

outline - універсальний атрибут, одночасно встановлює колір, стиль і товщину зовнішнього кордону на всіх чотирьох сторонах елемента. На відміну від лінії, яка задається через border, атрибут outline не впливає на положення блоку і його ширину. Також може використовуватись із outline-color, outline-style, outline-width.

Шрифт

font - шрифт. Універсальний параметр, який дозволяє встановити одночасно декілька атрибутів стилю шрифту.

font-family - сама назва говорить за себе, Сімейство шрифтів. Встановлює шрифт, який буде опреділятися для відображення тексту.

font-size - встановлює розмір шрифту, який буде відображатись на сайті.

font-style - визначає написання шрифту - звичайне, курсивне або похиле. Коли для тексту встановлено курсивне або похиле накреслення, браузер звертається до системи для пошуку відповідного шрифту. Якщо заданий шрифт не знайдений, браузер використовує спеціальний алгоритм для імітації потрібного виду тексту.

font-variant - визначає, як потрібно представляти літери в рядку? Робити їх все прописними зменшеного розміру або залишити без змін. Такий спосіб зміни символів називається капітеллю.

font-weight - встановлює насиченість шрифту. Значення встановлюється від 100 до 900 з кроком 100. Також допустимо використання ключових слів.

Текст

direction - атрибути unicode-bidi і direction задають, як повинен розташовуватися текст використовуваної мови - зліва направо чи справа наліво

letter-spacing - визначає інтервал між символами в межах елемента. Браузери зазвичай встановлюють відстань між символами, виходячи з типу і виду шрифту, його розмірів і налаштувань операційної системи. Щоб змінити це значення і застосовується цей атрибут.

line-height - встановлює міжрядковий інтервал тексту, відлік ведеться від базової лінії шрифту. За звичайних обставин відстань між рядками залежить від виду та розміру шрифту і визначається браузером автоматично.

text-align- визначає горизонтальне вирівнювання тексту в межах елемента.

text-decoration - додає оформлення тексту у вигляді його підкреслення, перекреслення, лінії над текстом і миготіння. Одночасно можна застосувати більш одного стилю, перераховуючи значення через пробіл.

text-indent - встановлює величину відступу першого рядка блоку тексту (наприклад, для параграфа P). Впливу на всі інші рядки не виявляється.

text-transform - управляє перетворенням тексту елемента в заголовні або прописні символи. Коли значення відмінне від none, регістр вихідного тексту буде змінено.

unicode-bidi - атрибути unicode-bidi і direction задають, як повинен розташовуватися текст використовуваної мови - зліва направо чи справа наліво.

vertical-align - вирівнює елемент по вертикалі щодо свого батька або навколишнього тексту.

white-space -параметр white-space встановлює, як відображати пропуски (пробіли) між словами.

word-spacing - встановлює інтервал між словами. Якщо встановлено параметр вирівнювання justify, то атрибут word-spacing не діє, оскільки інтервал між словами буде встановлений примусово.

Списки

list-style - атрибут, що дозволяє одночасно задати стиль маркера, його положення, а також зображення, яке буде використовуватися в якості маркера.

list-style-image - встановлює адресу зображення, яке служить в якості маркера списку. Цей атрибут успадковується, тому для окремих елементів списку для відновлення маркера використовується значення none.

list-style-position - визначає, як буде розміщуватися маркер щодо тексту. Є два значення: outside - маркер винесений за кордон елемента списку і inside - маркер обтікає текстом.

list-style-type - змінює вигляд маркера для кожного елемента списку. Цей атрибут використовується тільки у випадку, коли значення властивості list-style-image встановлено як none. Маркери розрізняються для маркованого списку (тег UL) і нумерованого (тег OL).

Форматування

clear - параметр встановлює, з якого боку елемента заборонено його обтікання іншими елементами. Якщо встановлено обтікання елемента за допомогою параметра float, властивість clear скасовує її дію для зазначених сторін.

clip - параметр clip визначає область позиціонованого елемента, в якій буде показано його вміст. Все, що не поміщається в цю область, буде обрізане і стає невидимим. На даний момент єдиною доступною формою області - прямокутник.

display - багатоцільовий атрибут, який визначає, як елемент повинен бути показаний в документі.

float - визначає, по якій стороні буде вирівнюватися елемент, при цьому інші елементи будуть обтікати його з інших сторін.

height - встановлює висоту блокових або замінних елементів (до них, наприклад, відноситься тег IMG). Висота не включає товщину кордонів навколо елемента, значення відступів і полів.

`max-height` - встановлює максимальну висоту елемента. Значення висоти елемента буде обчислюватися в залежності від установлених параметрів `height`, `max-height` і `min-height`.

`max-width` - встановлює максимальну ширину елемента.

`min-height` - визначає мінімальну висоту елемента. Значення висоти елемента буде обчислюватися в залежності від установлених параметрів `height`, `max-height` і `min-height`.

`min-width` - визначає мінімальну ширину елемента. Якщо вікно браузера досягає заданої мінімальної ширини елемента, то його ширина залишається незмінною і з'являється горизонтальна смуга прокручування.

`overflow` - властивість `overflow` управляє відображенням змісту блокового елемента, якщо воно цілком не поміщається і виходить за область заданих розмірів.

`visibility` - призначений для відображення або приховування елемента, включаючи рамку навколо нього і фон. При приховуванні елемента, хоча він і стає не видимим, місце, яке займає елемент, залишається за ним.

`width` - встановлює ширину блокових або замінних елементів. Ширина не включає товщину кордонів навколо елемента, значення відступів і полів

`width` - встановлює ширину блокових або замінних елементів. Ширина не включає товщину кордонів навколо елемента, значення відступів і полів.

Позиціонування

`bottom` - встановлює положення нижнього краю вмісту елемента без урахування товщини рамок і відступів. Відлік координат залежить від параметра `position`, він зазвичай приймає значення `relative` (відносне положення) або `absolute` (абсолютне положення).

`left` - для позиційованої елемента визначає відстань від лівого краю батьківського елемента, не включаючи відступ, поле і ширину рамки, до лівого краю дочірнього елемента. Відлік координат залежить від значення параметра `position`.

`position` - встановлює спосіб позиціонування елемента щодо вікна браузера або інших об'єктів на веб-сторінці.

`right` - для позиційованої елемента визначає відстань від правого краю батьківського елемента, не включаючи відступ, поле і ширину рамки, до правого краю дочірнього елемента.

top - для позиційованої елемента визначає відстань від верхнього краю батьківського елемента, не включаючи відступ, поле і ширину рамки, до верхнього краю дочірнього елемента.

z-index - управляє розміщенням елементів по z-осі, що дозволяє накладати елементи один на одного.

Відступи

margin - встановлює величину відступу від кожного краю елемента. Відступом є простір від кордону поточного елемента до внутрішнього кордону його батьківського елемента.

margin-bottom - встановлює величину відступу від нижнього краю елемента.

margin-left - встановлює величину відступу від лівого краю елемента.

margin-right - визначає величину відступу від правого краю елемента.

margin-top - встановлює величину відступу від верхнього краю елемента.

Поля

padding - встановлює значення полів навколо вмісту елемента. Полем називається відстань від внутрішнього краю рамки елемента до уявного прямокутника, що обмежує його вміст.

padding-bottom - встановлює значення поля від нижнього краю вмісту елемента.

padding-left - встановлює значення поля від лівого краю вмісту елемента.

padding-right - визначає значення поля від правого краю вмісту елемента.

padding-top - визначає величину поля від верхнього краю вмісту елемента.

Таблиці

border-collapse - встановлює, як відображати межі навколо клітинок таблиці. Цей параметр відіграє роль, коли для осередків встановлена рамка, тоді в місці стику осередків вийде лінія

подвійної товщини. Додавання значення collapse змушує браузер аналізувати подібні місця в таблиці і прибирати в ній подвійні лінії.

border-spacing - визначає відстань між кордонами осередків в таблиці. Атрибут border-spacing не працює у випадку, коли для таблиці встановлено параметр border-collapse із значенням collapse.

caption-side - визначає положення заголовка таблиці, який задається за допомогою тега CAPTION, щодо самої таблиці. Параметр caption-side виводить заголовок до або після таблиці, а вирівнювання тексту по правому чи лівому краю встановлюється через атрибут text-align.

empty-cells - параметр задає відображення кордонів і фону в комірці, якщо в ній немає видимого змісту. При одночасному додаванні до таблиці параметра border-collapse із значенням collapse, атрибут empty-cells ігнорується.

table-layout - визначає, як браузери мають обчислювати висоту і ширину осередків таблиці, ґрунтуючись на її вміст.

Інтерфейс

cursor - встановлює форму курсора, коли він знаходиться в межах елемента. Вигляд курсору залежить від операційної системи і встановлених параметрів.

Важливість

! important - грає роль у тому випадку, коли користувачі підключають свою власну таблицю стилів. Якщо виникає суперечність, коли стиль автора сторінки і користувача для одного і того ж елемента не збігається, то ! Important дозволяє підвищити пріоритет стильового параметра.

Псевдоелементи

after - псевдоелемент, який використовується для виведення бажаного контенту після елемента, до якого він додається. Псевдоелемент after працює спільно з атрибутом content.

before - псевдоелемент before застосовується для відображення бажаного контенту до елемента, до якого він додається. before працює спільно з атрибутом content.

first-letter - псевдоелемент first-letter визначає стиль першого символу в тексті елемента, до якого додається. До цього псевдоелементу можуть застосовуватися тільки стильові атрибути, пов'язані з властивостями шрифту, полями, відступами, кордонами, кольором і тлом (фоном).

Псевдокласи

`active` - псевдоклас `active` визначає стиль для активного посилання. Активної посилання стає при натисканні на неї.

`first-child` - псевдоклас `first-child` додає стильове оформлення до першого дочірньому елементу селектора, який розташований у списку елементів документа.

`focus` - псевдоклас `focus` визначає стиль для тегу, який отримує фокус. Наприклад, їм може бути текстове поле форми, в яке встановлюється курсор.

`hover` - визначає стиль тегу при наведенні на нього курсору миші.

`link` - псевдоклас `link` застосовується до посилань, які ще не відвідував користувач, і задає для них стильове оформлення.

`visited` - псевдоклас `visited` застосовується до посилань, вже відвідані користувачем, і задає для них стильове оформлення.

Контент

`content` - дозволяє вставляти генерується утримання в текст веб-сторінки, що спочатку в тексті відсутній.

`counter-increment` - призначений для збільшення значення лічильника збільшень, який задається параметром `counter-reset`. Такий лічильник підраховує кількість відображень елементів на сторінці та може виводитися за допомогою властивості `content` і псевдоелементів `after` і `before`. Це дозволяє створювати списки (у тому числі багаторівневі), в яких нумерація і вид задаються через стилі.

`counter-reset` - встановлює ідентифікатор, в якому буде зберігатися лічильник відображень певного елемента, а також початкове значення лічильника. Такий лічильник може виводитися за допомогою властивості `content` і псевдоелементів `after` і `before`.

