



Лекція

2

«Технологія CSS»

«Технологія

CSS»

План лекції

1. Основи CSS
2. CSS текст
3. CSS шрифти
4. CSS фон
5. CSS



1. ОСНОВЫ CSS

- **CSS (Cascading Style Sheets - каскадні таблиці стилів, вимовляється «сі ес-ес»)** - технологія управління зовнішнім виглядом тегів веб-сторінки



- CSS або каскадні таблиці описують правила форматування окремого елемента веб-сторінки. Створивши стиль один раз, його можна застосовувати до будь-яких елементів сторінки.
- Стилi є набором параметрів, що управляють видом і станом елементів веб-сторінки.

1.1. Переваги

використання СТИЛІВ



Розмежування коду та оформлення

- Ідея про те, щоб код HTML був вільний від елементів оформлення таких, як визначення кольору, розміру шрифту та інших параметрів, стара як світ.
- В ідеалі, веб-сторінка повинна містити тільки теги логічного форматування, а вигляд елементів задається через стилі.
- При подібному поділі робота над дизайном і версткою сайту може вестися паралельно.

Різне оформлення для різних пристроїв

- За допомогою стилів можна визначити вид веб-сторінки для різних пристроїв виводу: монітора, принтера, смартфона, планшета.
- Наприклад, на екрані монітора відобразити сторінку в одному оформленні, а при її друку - в іншому.
- Ця можливість також дозволяє приховувати або показувати деякі елементи або відображенні на різних документа при пристроях.

Способи оформлення елементів

- На відміну від HTML стилі мають набагато більше можливостей по оформленню елементів веб-сторінок. Простими засобами можна:
- змінити колір фону елемента;
- додати рамку;
- встановити шрифт;
- визначити розміри, положення тощо.

Прискорення завантаження сайту

- При зберіганні стилів в окремому файлі, він кешується і при повторному зверненні до нього витягується з кешу браузера. За рахунок кешування і того, що стилі зберігаються в окремому файлі, зменшується код веб-сторінок і **знижується час завантаження документів.**
- *Кешем називається спеціальне місце на локальному комп'ютері користувача, куди браузер зберігає файли при першому зверненні до сайту. При наступному зверненні до сайту ці файли вже не викачуються по мережі, а беруться з локального диска. Такий підхід дозволяє істотно підвищити швидкість завантаження веб-сторінок.*

Єдине стильове оформлення безлічі документів

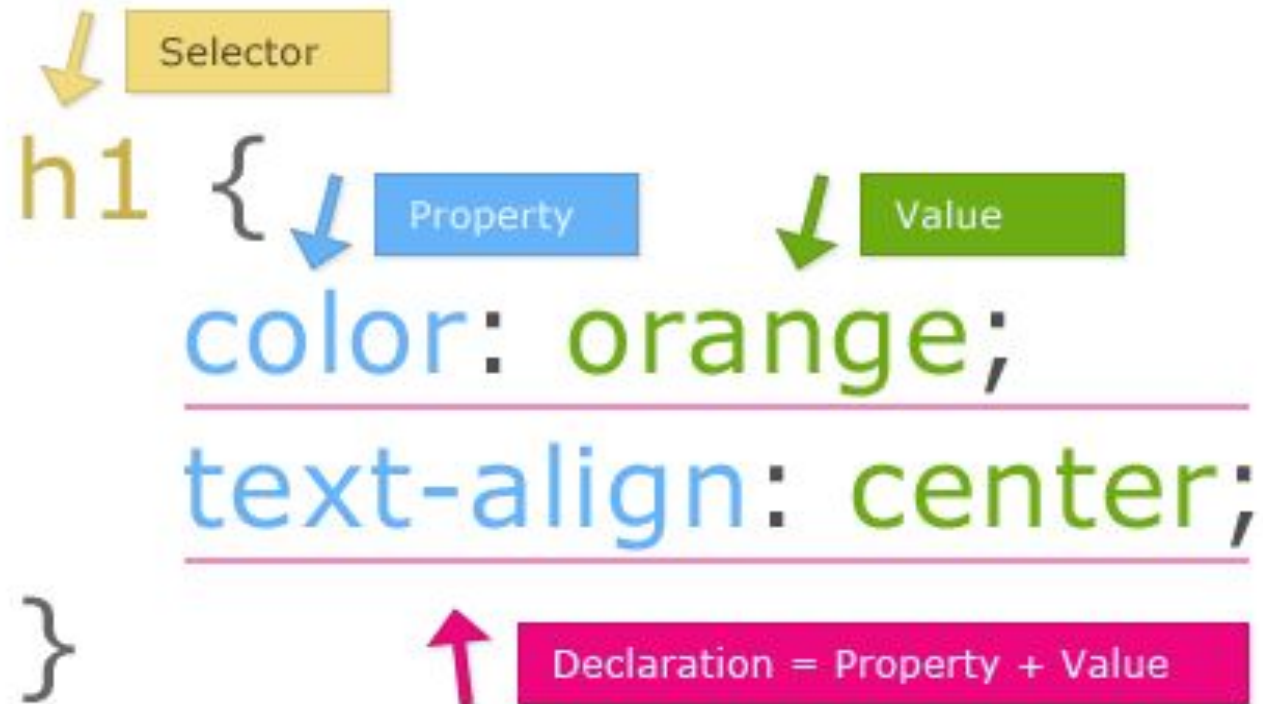
- Сайт - це не просто набір пов'язаних між собою документів, а й однакове розташування основних блоків та їх вигляд.
- Застосування однакового оформлення заголовків основного тексту та інших елементів створює спадкоємність між сторінками і є користувачам роботу з сайтом і його сприйняття в цілому. Розробникам же використання стилів істотно спрощує проектування дизайну.

Централізоване зберігання

Стилі, як правило, зберігаються в одному або декількох спеціальних файлах, посилання на які вказується в усіх документах сайту. Завдяки цьому зручно правити стиль в одному місці, при цьому оформлення елементів автоматично змінюється на всіх сторінках, які пов'язані із зазначеним файлом. Замість того щоб модифікувати десятки HTML-файлів, досить відредагувати один файл зі стилем і оформлення потрібних документів відразу ж зміниться.

1.2. Синтаксис CSS

Anatomy of a CSS Rule



Базовий синтаксис CSS



- 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 перераховуються команди.
- CSS не чутливий до регістру, перенесення рядків, пробілів і символів табуляції, тому форма запису залежить від бажання розробника.

Прикла

Д

index.htm

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Introducing CSS</title>
5   <link href="css/example.css" type="text/css" rel="stylesheet" />
6 </head>
7 <body>
8   <h1>From Garden to Plate</h1>
9   <p>A <i>potager</i> is a French term for an ornamental vegetable or kitchen garden ... </p>
10  <h2>What to Plant</h2>
11  <p>Plants are chosen as much for their functionality as for their color and form ... </p>
12 </body>
13 </html>
```

example.css

```
1 body {
2   font-family: Arial, Verdana, sans-serif;
3 }
4 h1, h2 {
5   color: #ee3e80;
6 }
7 p {
8   color: #665544;
9 }
```

From Garden to Plate

A *potager* is a French term for an ornamental vegetable or kitchen garden ...

What to Plant

Plants are chosen as much for their functionality as for their color and form ...

Форма запису

- Для селектора допускається додавати кожну стильову властивість і її значення окремо, як це показано в прикладі.

```
td { background: olive; }  
td { color: white; }  
td { border: 1px solid black; }
```

- Однак такий запис не дуже зручний. Доводиться повторювати кілька разів один і той же селектор, та й легко заплутатися в їх

```
td {  
  background:  
  olive; color:  
  white;  
  border: 1px solid black;  
}
```


Пріоритет застосування стилів

- Якщо для селектора спочатку задається властивість з одним значенням, а потім та ж властивість, але вже з іншим значенням, то застосовуватися буде те значення, яке в коді встановлено нижче.
- *Пр*

```
p { color: green; }  
p { color: red; }
```
- В даному прикладі для селектора p колір тексту спочатку встановлений зеленим, а потім червоним. Оскільки значення red розташоване нижче, то воно в підсумку і буде застосовуватися до тексту.
- Насправді такого запису краще взагалі уникати і видаляти значення, що повторюються. Але подібне може статися випадково, наприклад, в разі підключення різних стильових файлів, в яких

Значенн я

- У кожної властивості може бути тільки відповідне її функції значення.
- Наприклад, для `color`, який встановлює колір тексту, як значення неприпустимо використовувати числа.

Коментарі

- **Коментарі** потрібні, щоб робити пояснення з приводу використання стильових властивостей, виділяти розділи або писати свої нотатки. Коментарі дозволяють легко згадувати логіку і структуру селектора і підвищують розбірливість коду.
- Разом з тим, додавання тексту збільшує обсяг документів, що негативно позначається на часі їх завантаження. Тому коментарі, зазвичай, застосовують в налагоджувальних або навчальних цілях, щоб помітити, що текст є коментарем, а при викладанні сайту в мережу їх видаляють наступну конструкцію `/* ... */`.

- ```
/*
Стиль для сайта htmlbook.ru
Сделан для ознакомительных целей
*/
```

- Прикла

```
div {
 width: 200px; /* Ширина блоку */
 margin: 10px; /* Поля навколо елемента */
 float: left; /* Обгорнення по правому краю
 */
}
```

# 1.3. Способи додавання стилів на сторінку



# 1. Вбудовані таблиці стилів INTERNAL CSS

- **Вбудовувані стилі** – це набори стилів, які є частиною коду веб-сторінки, вкладені між тегами `<style> ... </style>` і розташовані усередині елемента `<head>`.
- Вбудовувані стилі діють тільки на сторінці, на якій вони містяться.
- На одній сторінці можна розміщувати будь-яку кількість

# Приклад вбудованих СТИПІВ

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Using Internal CSS</title>
5 <style type="text/css">
6 body {
7 font-family: arial;
8 background-color: rgb(185,179,175);
9 }
10 h1 {
11 color: rgb(255,255,255);
12 }
13 </style>
14 </head>
15 <body>
16 <h1>Potatoes</h1>
17 <p>There are dozens of different potato varieties.
18 They are usually described as early, second early and
19 maincrop.</p>
20 </body>
21 </html>
```

## Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop.

## 2. Зовнішні таблиці стилів EXTERNAL CSS

- **Зовнішня таблиця стилів** – це текстовий файл, в якому знаходиться весь набір стилів CSS.
- Назва цього файлу завжди має закінчуватися розширенням `.css`. Задані таким чином стилі працюватимуть для всіх сторінок веб-сайту, до яких вони підключені.
- До веб-сторінки можна приєднати кілька таблиць стилів, додаючи послідовно кілька тегів `<link>`, вказавши в атрибуті тега `href` шлях до даної таблиці стилів.



## 3. Зовнішні таблиці стилів

- Зовнішня таблиця стилів прикріплюється до веб-сторінки за допомогою html-тега `<link>`, вкладеного в тег `<head>`:

```
<head>
 <link href="css/styles.css" type="text/css" rel="stylesheet">
</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).

# Приклад зовнішніх таблиць стилів

index.htm

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Using External CSS</title>
5 <link href="css/styles.css" type="text/css" rel="stylesheet" />
6 </head>
7 <body>
8 <h1>Potatoes</h1>
9 <p>There are dozens of different potato varieties. They
10 are usually described as early, second early and maincrop.</p>
11 </body>
12 </html>
```

styles.c

```
1 body {
2 font-family: arial;
3 background-color: rgb(185,179,175);
4 }
5 h1 {
6 color: rgb(255,255,255);
7 }
```

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop.

# 3. Внутрішньотекстові таблиці стилів (інлайнові стилі)

- Внутрішньотекстові стилі не використовують селектори, їх застосування

постигає в прикладі стилів безпосередньо до HTML-елементу через атрибут `style`.

```
<p style="font-family: 'Times New Roman', Georgia, Serif; color: #70d7700;">
```

```
Обратите внимание на этот текст.</p>
```

- Недолік цього способу полягає в відсутності можливості автоматичного використання даного стилю для іншого елемента.
- *Внутрішні стилі рекомендується застосовувати на сайті обмежено або взагалі відмовитися від їх використання. Справа в тому, що додавання таких стилів збільшує загальний обсяг файлів, що веде до підвищення часу їх завантаження в браузері, і ускладнює редагування документів для розробників.*

# 4. Правило @import

- Правило @import дозволяє завантажити зовнішню таблицю стилів.
- Щоб директива @import працювала, вона повинна розташовуватися всередині тега <style> і перед всіма іншими

```
@import url("і'мя файлу") [типи носіїв];
@import "і'мя файлу" [типи носіїв];
```

# Типи носіїв та їх ОПИС

Тип	Опис
<b>all</b>	Всі типи. Це значення використовується за умовчанням
<b>aural</b>	Мовні синтезатори, а також програми для відтворення тексту вголос. Сюди, наприклад, можна віднести мовні браузер
<b>braille</b>	Пристрої, засновані на системі Брайля, які призначені для сліпих людей
<b>handheld</b>	Надолонні комп'ютери і аналогічні їм апарати
<b>print</b>	Друкуючі пристрої на кшталт принтера
<b>projection</b>	Проектор
<b>screen</b>	Екран монітора
<b>tv</b>	Телевізор

- Використання типів носіїв спільно з імпортом файлу дає можливість вказувати стиль тільки для певних пристроїв.
- Як значення використовується шлях до стильового файлу, який вказується всередині необов'язковою конструкції url (). Шлях до файлу при цьому можна писати як в папках (подвійних або одинарних) так і без НИХ.

- *Приклад 1:*

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Импорт стиля</title>
 <style>
 @import "/style/main.css" screen; /* Стиль для
вывода результата на монитор */
 @import "/style/palm.css" handheld, print; /* Стиль для
печати
*/
 </style>
 </head>
 <body>
 <p>...</p>
 </body>
</html>
```

- *Приклад*  
2:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
...
<style type="text/css">
@import url(mobile.css);
p {
font-size: 0.9em;
color: grey;
}
</style>
...
```

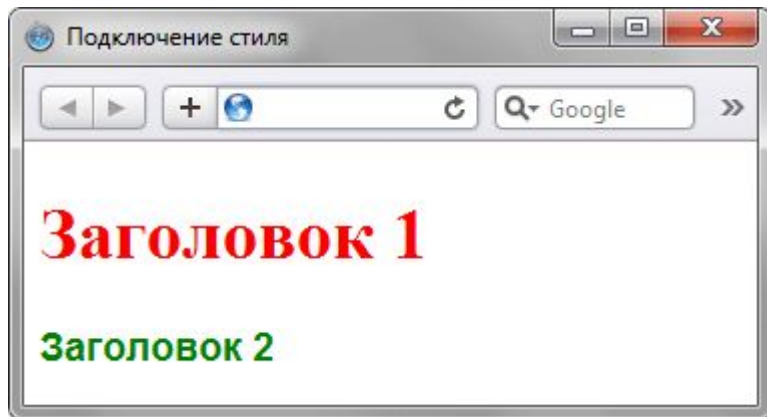
# Комплексне використання стилів

- Всі описані методи використання CSS можуть застосовуватися як самотійно, так і в поєднанні один з одним. В цьому випадку необхідно пам'ятати про їх ієрархії.
- Першим має пріоритет інлайновий стиль, потім вбудований стиль і в останню чергу зовнішні таблиці стилів.
- У прикладі застосовується відразу два методи додавання стилю в документ.



- *Приклад*

:



```
<!DOCTYPE HTML>
<html>
 <head>
 <meta charset="utf-8">
 <title>Подключение стиля</title>
 <style>
 H1 {
 font-size: 120%;
 font-family: Arial, Helvetica, sans-serif;
 color: green;
 }
 </style>
 </head>
 <body>
 <h1 style="font-size: 36px; font-family: Times, serif; color:
red">Заголовок 1</h1>
 <h1>Заголовок 2</h1>
 </body>
</html>
```

## 1.4. Принципи концепції CSS

# Успадковуван НЯ

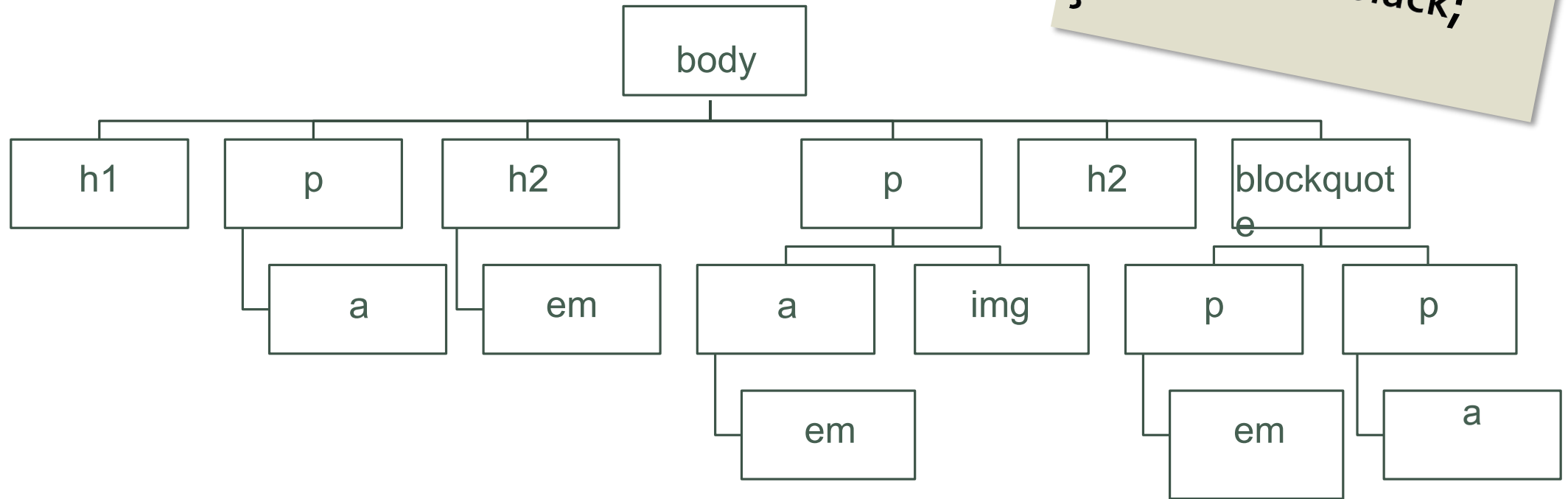
- Почнемо з найпростішої для розуміння концепції CSS. Суть її полягає в тому, що стилі, присвоєні деякому елементу, успадковуються всіма нащадками (вкладеними елементами), якщо вони не перевизначені явно.
- Наприклад, розмір шрифту і його колір досить застосувати до дескриптора **body**, щоб всі елементи всередині мали ті ж властивості.
- Але, для заголовків h1-h6 розмір шрифту не буде присвоєний, тому що у браузера для них є своя таблиця стилів за замовчуванням, а успадковані стилі мають найнижчий пріоритет. Аналогічна ситуація з кольором посилань.



Як ви думаєте, хто  
успадкує  
ознаку?

Відзначте всіх нащадків, які успадковують зелений колір елемента `<body>`. Але спочатку погляньте на CSS-код, наведений вище.

```
body {
 color: green;
}
p {
 color: black;
}
```



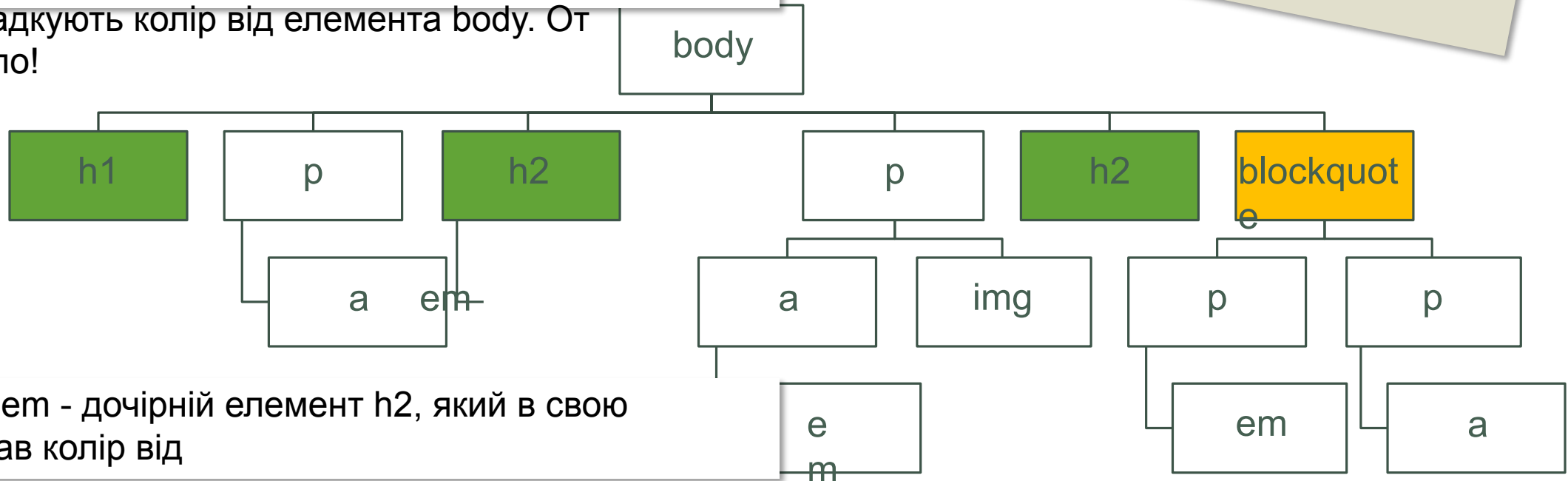
# Хто успадковує ознаки?

Для елемента `blockquote` немає CSS-правила, тому він теж успадкує колір від елемента `body`. Однак оскільки селектор `p` змінює колір на чорний, `blockquote` не стане зеленим

Оскільки у елементів `h1` і `h2` немає своєї властивості `color`, вони успадкують колір від елемента `body`. От пощастило!

```
body {
 color: green;
}

p {
 color: black;
}
```



Елемент `em` - дочірній елемент `h2`, який в свою успадкував колір від `body`

# 1.5. Основні типи

селекторів CSS

- За допомогою **селекторів** створюються CSS-правила для форматування елементів веб-сторінки. Як селектор використовуються не тільки самі елементи і їх класи, але також ідентифікатори, 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



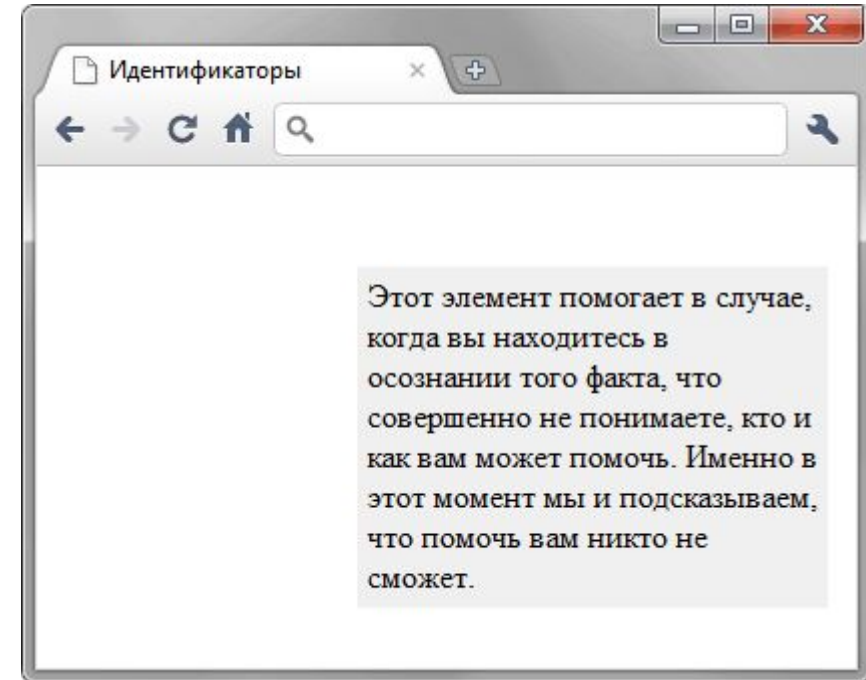
Селектор	Опис
<b>універсальний</b>	Універсальний селектор відповідає будь-якому елементу, наприклад, такий запис обнулить відступи для всіх елементів веб-сайту: <b>* {margin: 0;}</b>
<b>елемента</b>	Селектори елементів використовуються для визначення стилів елементів для всіх сторінок веб-сайту, наприклад, стиль заголовків h1 або загальний стиль абзаців: <b>h1 {font-family: Lobster, cursive;} p {letter-spacing: 0.1em;}</b>

Селектор	Опис
<b>класу</b>	<p>Селектори класу використовуються для визначення стилів для декількох елементів одного типу, розміщених в різних частинах або на різних сторінках веб-сайту. Наприклад, для створення заголовка класу <code>headline</code> необхідно додати атрибут <code>class</code> з відповідним значенням в відкриваючий тег <code>&lt;h1&gt;</code>. Далі необхідно задати стиль для зазначеного класу. Даний стиль оформлення можна застосувати і для інших елементів.</p> <p><i>Код HTML:</i></p> <pre><b>&lt;h1 class="headline"&gt;Методичні вказівки&lt;/h1&gt;</b></pre> <p><i>Код CSS:</i></p> <pre><b>.headline { text-transform: uppercase; color: lightblue;}</b></pre>

Селектор	Опис
ідентифікатора	<p>Селектори ідентифікатора використовуються для присвоювання стилю одному конкретному елементу. Ідентифікатор id елемента можна використовувати в документі тільки один раз, так як він виділяє унікальний елемент. <b>#sidebar { text-transform: uppercase; color: lightblue;}</b></p>

# Приклад використання селектора ідентифікатора

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Ідентифікатори</title>
 <style>
 #help {
 position: absolute; /* Абсолютне позиціонування */
 left: 160px; /* Положення елемента від лівого краю */ top:
 50px; /* Положення від верхнього краю */
 width: 225px; /* Ширина блоку */ padding:
 5px; /* Поля навколо тексту */ background:
 #f0f0f0; /* Колір фону */
 }
 </style>
 </head>
 <body>
 <div id="help">
 Этот элемент помогает в случае, когда вы находитесь в осознании
 того
 факта, что совершенно не понимаете, кто и как вам может помочь.
 Именно
 в этот момент мы и подсказываем, что помочь вам никто не сможет.
 </div>
 </body>
</html>
```



## Селектор

## Опис

### 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, які є нащадками всіх елементів ul.

Якщо потрібно відформатувати нащадки певного елемента, то можна задати 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 будь-якого посилання, розташованого усередині інших тегів, позначених класом

.first.

Селектор	Опис
<b>дочірній</b>	<p>Дочірній тег є прямим нащадком тега, в якому він міститься, тобто відносини «батьки-діти» існують між елементами і тими елементами, які містяться безпосередньо всередині них. У одного елемента може бути кілька дочірніх елементів, а батьківський елемент може бути у кожного елемента тільки один.</p> <p><b>p &gt; strong</b> - вибирає всі елементи strong, які є дочірніми по відношенню до елементу p</p>

Селектор	Описание
сестринський	<p>Сестринські відносини виникають між елементами, що мають спільного батька. Селектори сестринських елементів дозволяють вибрати теги з групи елементів одного рівня.</p> <p><b>h1 + p</b> - вибере всі перші абзаци, що йдуть безпосередньо за будь-яким тегом &lt;h1&gt;, не зачіпаючи інші абзаци.</p> <p><b>h1 ~ p</b> - вибере всі абзаци, які є сестринськими по відношенню до будь-якого заголовку h1 і йдуть після нього.</p>

# Приклад використання сестринського селектора

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>Соседние селекторы</title>
 <style>
 b + i {
 color: red; /* Красный цвет текста */
 }
 </style>
 </head>
 <body>
 <p>Lorem ipsum dolor sit amet, <i>consectetuer</i> adipiscing elit.</p>
 <p>Lorem ipsum dolor sit amet, <i>consectetuer</i> adipiscing elit.</p>
 </body>
</html>
```





# Прикла

Д

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Using External CSS</title>
5 <link href="css/styles.css" type="text/css" rel="stylesheet" />
6 </head>
7 <body>
8 <h1>Potatoes</h1>
9 <p id="intro">There are <i>dozens</i> of different potato varieties.</p>
10 <p>They are usually described as early, second early and maincrop potatoes.</p>
11 </body>
12 </html>
```

```
1 * { font-family: Arial, Verdana, sans-serif;}
2 h1 { font-family: "Courier New", monospace;}
3 i { color: green;}
4 i { color: red;}
5 b { color: pink;}
6 p b { color: blue !important;}
7 p b { color: violet;}
8 p#intro { font-size: 100%;}
9 p { font-size: 75%;}
```

## Potatoes

There are *dozens* of different **potato** varieties.

They are usually described as early, second early and maincrop potatoes.

# 2. CSS Текст



- **CSS текст** - це набір css-стилів для форматування текстового вмісту веб-сторінок.
- Використання CSS-стилів для форматування тексту дозволяє надати html-елементам бажаний вид, завдяки чому html-теги можуть застосовуватися тільки за своїм прямим призначенням - для визначення структури документа.

## 2.1. Горизонтальне вирівнювання `text-align`

- Властивість визначає, як будуть вирівнюватися рядки тексту елемента щодо меж блоку.
- Застосовується тільки до блокових елементів, наприклад, абзаців.

## Значення: `text-align`

### **left**

Вирівнювання по лівому краю елемента. Синтаксис: `p {text-align: left;}`

### **right**

Вирівнювання по правому краю елемента. Синтаксис: `p {text-align: right;}`

### **center**

Вирівнювання по центру елемента, управляє вирівнюванням вмісту, а не самих елементів. Центрує кожен рядок тексту елемента. Синтаксис: `p {text-align: center;}`

### **justify**

Вирівнювання по ширині елемента. У вирівняному по ширині тексті обидва кінці рядка притискаються до лівого і правому краях батьківського елемента. Прогалини між словами і буквами розподіляються таким чином, щоб довжина всіх рядків дорівнювала. Різні браузері можуть збільшити як відступи між словами, так і інтервали між буквами. Синтаксис: `p {text-align: justify;}`

## 2.2. Відступ text-indent

- Властивість встановлює відступ (виступ) в першому рядку елемента, допомагаючи створити ілюзію структурованого тексту. У загальному випадку можна застосовувати до будь-якого блочному елементу, задаючи як позитивні, так і негативні значення.
- Якщо в першому рядку блочного елемента присутнє зображення, то воно зрушиться разом з іншим текстом.

# text-indent

Значенн

я:

**Довжин  
а  
(em,px)**

Зміщує перший рядок на задану величину.  
Можна задавати як позитивні, так і негативні  
значення Синтаксис: **p {text-indent: 5px;}**

**%**

Величина зміщення задається у  
відсотках. Синтаксис: **p {text-indent:  
2%;}**

# Приклад реалізації властивості `text-indent`

Отступ первой строки этого абзаца задан с помощью свойства `text-indent: 25px`.  
Остальные строки абзаца не будут иметь отступ, независимо от длины абзаца.

Для этого абзаца задан выступ (отрицательный отступ) первой строки `text-indent: -25px`.  
Остальные строки абзаца не будут иметь отступ, независимо от длины абзаца.



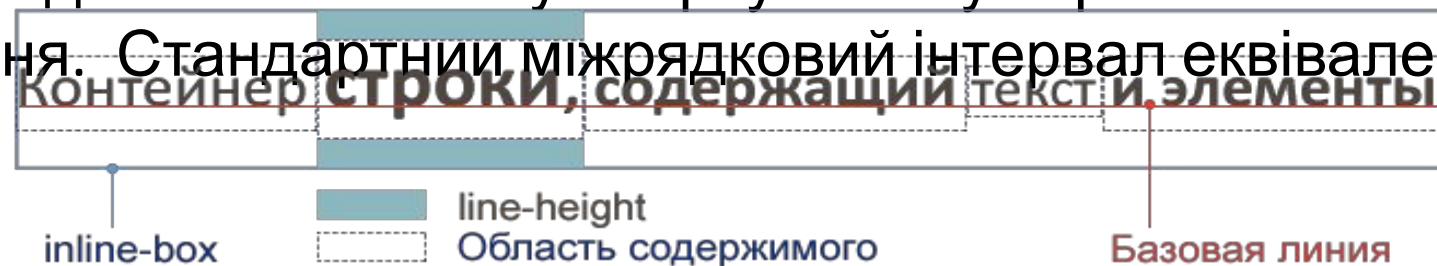
Отрицательный отступ `text-indent: -30px` также можно использовать для выделения первой строки абзаца с помощью изображений.



## 2.3. Вертикальне вирівнювання

### line-height

- **Висота рядків.** Властивість задає відстань між базовими лініями рядків тексту, визначаючи величину, на яку збільшується або зменшується висота блоку кожного елемента (управління міжрядковим інтервалом).
- Щоб визначити міжрядковий інтервал, потрібно знайти різницю між line-height і font-size, різницю поділити на два, а кожену половину додати до області вмісту зверху і знизу. Приймає тільки позитивні значення. Стандартний міжрядковий інтервал еквівалентний 120%.



# line-height

Значення:

**Довжина**  
(px,pt)

Фіксоване значення висоти рядка. Синтаксис: **h1 {line-height: 20px;}**

**Відносна**  
**значення**  
(em, ex,  
%)

Обчислюється щодо значення властивості font-size. Синтаксис: **h1 {line-height: 200%;}**

**Число**

Щоб установити величину міжрядкового інтервалу використовується просте числове значення, наприклад line-height: 1,5;. При заданні числа без одиниці вимірювання успадковується коефіцієнт масштабування, тобто line-height кожного елемента буде обчислюватися щодо його значення font-size.

Синтаксис: **h1 {line-height: 1.2;}**

**normal**

Значення браузера за замовчуванням. Зазвичай в 1,2 рази більше розміру шрифту. Синтаксис: **h1 {line-height: normal;}**

# Приклад реалізації міжрядкових інтервалів за допомогою `line-height`

Межстрочный интервал позволяет управлять расстоянием между строками текста. В данном примере при размере шрифта `{font-size: 1em}` задан межстрочный интервал `{line-height: 0.8em}`.

Для данного абзаца при размере шрифта `{font-size: 1em}` задан межстрочный интервал `{line-height: 1}`.

В данном абзаце при размере шрифта `{font-size: 1em}` установлен межстрочный интервал `{line-height: normal}`.

Оптимальным значением межстрочного интервала при размере шрифта `{font-size: 1em}` будет `{line-height: 1.4em}`.

## 2.4. Вертикальне вирівнювання vertical-align

- Застосовується тільки до рядкових елементів, до зображень і полів форм. НЕ вирівнює вміст блочного елемента та не успадковується.
- Позначається як vertical-align


# vertical-align

n

Значення	
Я: <b>baseline</b>	Вирівнює базову лінію елемента за базовою лінією його батька, поєднуючи середню лінію елемента із середньою лінією батьківського елемента. Синтаксис: <code>img {vertical-align: baseline;}</code>
<b>sub</b>	Робить елемент підстрочним (аналогічно з тегом <sub>). Величина зниження елемента може змінюватися в залежності від браузера користувача. Синтаксис: <code>img {vertical-align: sub;}</code>
<b>super</b>	Робить елемент надстрочним (аналогічно з тегом <sup>). При цьому значення sup і super не змінюють розмір шрифту, за замовчуванням текст надрядкового і підрядкового елемента має такий же розмір, як і текст батьківського елемента. Синтаксис: <code>img {vertical-align: super;}</code>
<b>top</b>	Верхній край елемента поєднується з верхнім краєм найвищого елемента в лінії.
<b>text-top</b>	Верхній край елемента поєднується з верхнім краєм шрифту батьківського елемента. Синтаксис: <code>img {vertical-align: text-top;}</code>
<b>middle</b>	Середня лінія елемента (зазвичай зображення) поєднується з лінією, що проходить через середину
<b>bottom</b>	Нижній край елемента поєднується з нижнім краєм найнижчого елемента в лінії. Синтаксис: <code>img {vertical-align: bottom;}</code>
<b>text-bottom</b>	Нижній край елемента поєднується з нижнім краєм шрифту батьківського елемента. Синтаксис: <code>img {vertical-align: text-bottom;}</code>
<b>длина (px,em)</b>	Дозволяє перемістити елемент на задану відстань. Синтаксис: <code>img {vertical-align: 5px;}</code>
<b>%</b>	Не дозволяє встановлювати middle, обчислюється як частина line-height елемента, а не його батька, тобто якщо встановити значення vertical-align, рівне 50% для елемента з line-height равним 20px, то базова лінія елемента

# Приклад реалізації властивості

## `vertical-align`

Изображение  `{vertical-align: baseline}` не имеет базовой линии, поэтому оно выравнивается по базовой линии родителя (в данном случае, текста).

Этот абзац содержит `надстрочный {vertical-align: super}` и `подстрочный {vertical-align: sub}` текст, который имеет такой же размер, как и основной текст.





Изображение  `{vertical-align: bottom}` выравнивается по низу контейнера строки, смещаясь ниже базовой линии текста.




Изображение  `{vertical-align: text-bottom}` выравнивается относительно нижней линии текста строки.



Изображение  `{vertical-align: top}` выравнивается по верху контейнера строки, при этом учитывается размер шрифта родительского элемента.

В данном примере изображение  `{vertical-align: text-top}` выравнивается относительно верхней линии текста строки.



Значение  `{vertical-align: middle}` выравнивает середину изображения по точке, которая находится на расстоянии, равном половине размера шрифта родительского элемента, над базовой линией текста.



# 3. CSS

## Шрифт

*(Типографік  
а)*

- **CSS шрифти** керують зовнішнім виглядом шрифту тексту веб-сторінок. Використовуючи різні шрифти для заголовків, абзаців та інших елементів, можна задавати певний стиль письмових повідомлень, передаючи бажані емоції і настрої, тому до вибору шрифтів для своїх сторінок потрібно підходити дуже виважено, продумано застосовуючи до них стилі.
- Не рекомендується використовувати більше двох шрифтів на сторінці, а бажаного контрасту можна досягти за рахунок комбінування шрифтів різної товщини, розміру, накреслення або ж за допомогою кольору.



# Форматування шрифту за допомогою CSS- властивостей

- Сімейство шрифтів **font-family**
- Стиль шрифту **font-style**
- Варіанти шрифтів **font-variant**
- Насиченість шрифту **font-weight**
- Розмір шрифту **font-size**
- Колір шрифту **color**
- Короткий список властивостей шрифту

# Свій шрифт на сторінці

- Правило `@ font-face` дозволяє визначити настройки шрифтів, а також завантажити специфічний шрифт на комп'ютер користувача.

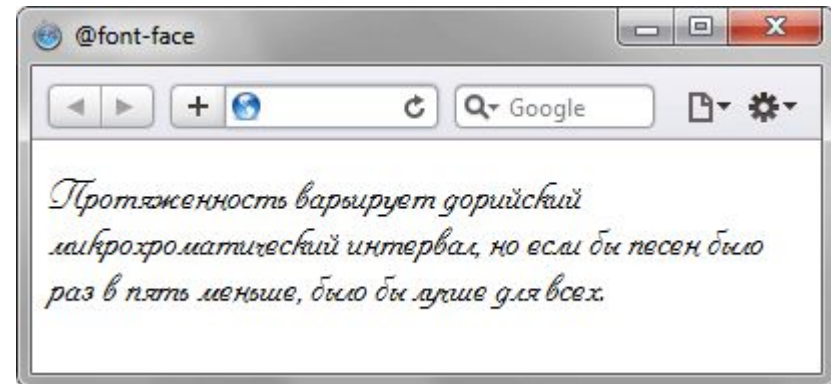
Синтаксис

```
@font-face { свойства шрифта
```

- Всередині конструкції `@ font-face` може перебувати набір властивостей для зміни параметрів шрифту (`font-family`, `font-size`, `font-style` і ін.), а також посилання на шрифтової файл.
- Посилання записується у вигляді `src: url (URI)`,

# Свій шрифт на сторінці (приклад)

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <title>@font-face</title>
 <style>
 @font-face {
 font-family: Pompadur; /* Имя шрифта */
 src: url(fonts/pompadur.ttf); /* Путь к файлу со шрифтом */
 }
 P {
 font-family: Pompadur;
 }
 </style>
 </head>
 <body>
 <p>Протяженность варьирует дорийский микрохроматический интервал,
 но если бы песен было раз в пять меньше, было бы лучше для всех.</p>
 </body>
</html>
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Font Family</title>
5 <style type="text/css">
6 body { font-family: Georgia, Times, serif;}
7 h1, h2 { font-family: Arial, Verdana, sans-serif;}
8 .credits { font-family: "Courier New", Courier, monospace;}
9 </style>
10 </head>
11 <body>
12 <h1>Briards</h1>
13 <p class="credits">by Ivy Duckett</p>
14 <p class="intro">The
15 briard,
16 or berger de brie, is a large breed of dog traditionally used as a herder
17 and guardian of sheep...</p>
18 </body>
19 </html>
```

# Прикла Д



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Font Family</title>
5 <style type="text/css">
6 body { font-family: Arial, Verdana, sans-serif;
7 font-size: 12px;}
8 h1 { font-size: 200%;}
9 h2 { font-size: 1.3em;}
10 </style>
11 </head>
12 <body>
13 <h1>Briards</h1>
14 <p class="credits">by Ivy Duckett</p>
15 <p class="intro">The
16 briard,
17 or berger de brie, is a large breed of dog traditionally used as a herder
18 and guardian of sheep...</p>
19 <h2>Breed History</h2>
20 <p>Lorem ipsum dolor sit amet, consectetur.
21 </body>
22 </html>
```

# Прикла

# Д

## Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep...

### Breed History

Lorem ipsum dolor sit amet, consectetur.

- Форматування шрифту за допомогою CSS-властивостей
- <http://html5book.ru/css-shrifty/>
- Обираємо шрифт для сайту
- <http://html5book.ru/shrift-dlya-sita/>

# 4. CSS ΦΟΗ

# CSS

## фон

- Дозволяє за допомогою css-властивості **background** додавати фон для будь-якого html-елемента. Кожна веб-сторінка має фон, на якому можна розміщувати малюнки і текст. В якості фону будь-якого елемента можна задавати колір, градієнт або зображення.
- Щоб встановити фон для одного елемента з групи, потрібно призначити для цього елемента клас (наприклад, `<div class = "header_box"> </div>`), ідентифікатор (наприклад, `<div id = "footer_box"> </ div>`) , або скористатися при відборі селекторами.
- Властивості фону не успадковуються, але фон батьківського блоку завжди буде видно.
- Деякі браузері відображають фон HTML-документів, заданий для елемента `<html>`. але рекомендується встановлювати фон для



# Фон і фонові зображення на веб-сторінці

- Колір зображення **background-color**
- Фонове зображення **background-image**
- Повтор фонових зображень **background-repeat**
- Позиціонування фонових зображень **background-position**
- Фіксація зображення на місці **background-attachment**
- Заповнення фоном відступів і кордонів елемента **background-clip**
- Положення фонового зображення щодо його батьківського блоку **background-origin**
- Розмір зображення **background-size**

# 5. CSS

## Посилання



# Псевдокласи станів гіпертекстових посилань

Більшість браузерів виділяють чотири основні стани гіперпосилань, кожному з яких відповідає свій **псевдоклас селектора**:

- Невідвідана — **a:link**
- Відвідана — по який вже здійснили перехід — **a:visited**
- Ненатиснута — над якою знаходиться вказівник миші — **a:hover**

# Прикла

Д

```
a:link {
 color: #497DDD;
 border-bottom: 1px dashed;
}
a:visited {
 color: #EF7D55;
}
a:hover {
 color: #154088;
 border-bottom: .07em solid;
}
a:active {
 color: #497DDD;
 border-bottom: 1px dashed;
}
```

# Стилізація посилань

- Для стилізації окремих посилань потрібно задати їм стильовий клас, після чого можна буде міняти зовнішній вигляд обраних посилань:

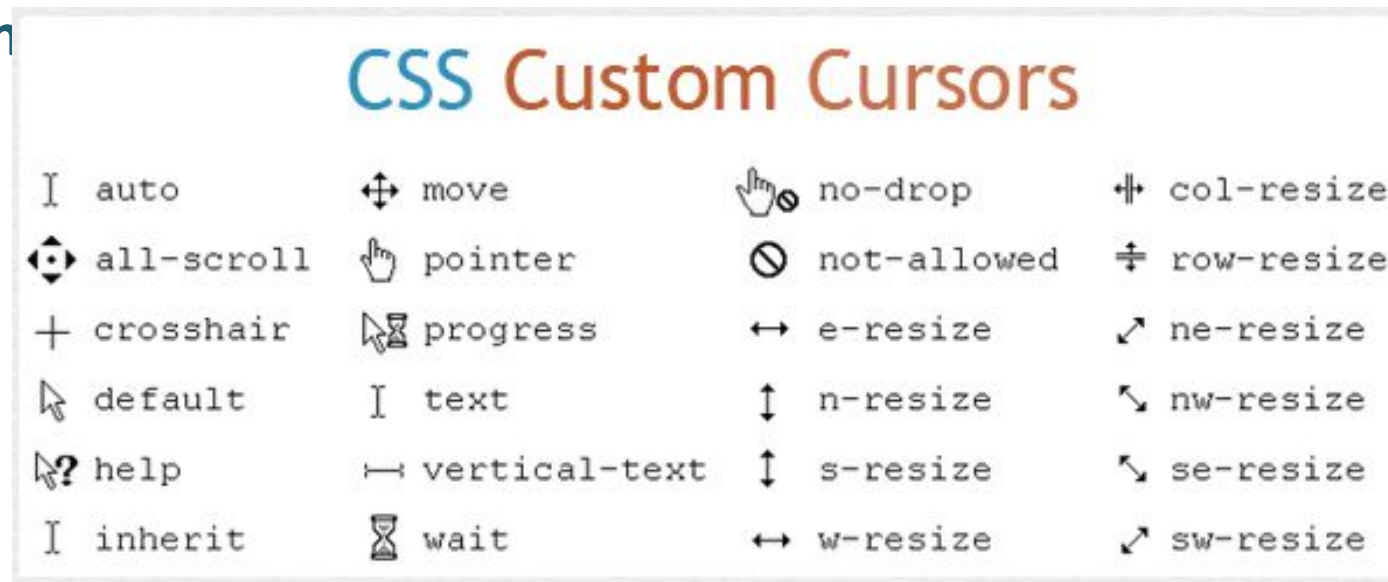
```
какой-то текст
```

- Прибираємо підкреслення:

```
a {
 text-decoration: none;
 border-bottom: 2px dashed DarkOrchid;
 padding-bottom: 3px;
}
```

# Зовнішній вид курсора миші

- Курсор миші МОЖ мати вигляд тако МОЖН встановит користувачьке зображення в якості курсора Наведіть над елементами таблиці нижче, щоб побачити, як виглядає курсор для кожного встановленого значення. Значення за замовчуванням `cursor: pointer`



# Література

