

Лекція 4.

СЕМАНТИЧНА СТРУКТУРА СТОРІНКИ HTML

4.1. Основні розділи документа

Веб-сторінки можуть і будуть виглядати досить відмінними одна від одної, але всі вони, як правило, мають спільні стандартні компоненти, якщо тільки сторінка не відображає повноекранне відео або гру, або не є частиною якогось арт-проекту, або просто погано структурована:

заголовок:

Зазвичай велика смуга поперек верху з великим заголовком, логотипом, і, можливо, хештегом. Заголовок зазвичай залишається однаковим від однієї веб-сторінки сайту до іншої.

панель навігації:

Посилання на основні розділи сайту; зазвичай, представлені кнопками меню, посиланнями або вкладками. Як і заголовок, цей контент зазвичай залишається однаковим для всіх веб-сторінок сайту – непослідовна навігація на сайті призведе до заплутаних та розчарованих користувачів. Багато веб-дизайнерів вважають навігаційну панель частиною заголовка, а не окремим компонентом, але це не є обов'язковою вимогою. Деякі дизайнери також стверджують, що наявність окремо навігаційної панелі та заголовка краще для доступності, оскільки переглядачі екрану можуть читати окремо функції навігації, якщо вони є окремою частиною.

основний зміст:

Велика область в центрі, яка містить більшу частину унікального вмісту даної веб-сторінки, наприклад, відео, яке ви хочете переглянути, або головну історію, яку ви читаєте, або карту, яку ви хочете переглянути, або заголовки новин і т.д. це єдина частина сайту, яка точно буде відрізнятися від однієї сторінки сайту до іншої.

бічна панель:

Ця панель містить деякі додаткові дані: посилання, цитати, оголошення тощо. Зазвичай, ці дані стосуються вмісту основного контенту (наприклад, на сторінці статей новин, бічна панель може містити біографію автора або посилання на пов'язані статті). Але в деяких випадках, тут можуть бути продубльовані елементи, такі як ще одна навігаційна система.

нижній колонтитул:

Смужка внизу сторінки, яка, як правило, містить повідомлення про авторські права або контактну інформацію. Це місце для розміщення загальної інформації (наприклад, заголовку), але, як правило, ця інформація не є критичною або є вторинною для самого веб-сайту. Нижній колонтитул також іноді використовується в цілях SEO, надаючи посилання для швидкого доступу до популярного контенту.

SEO (оптимізація пошукової системи) – це процес, який робить веб-сайт більш помітним в результатах пошуку, також відомий як *Підвищення пошукового рейтингу*.

Пошукові системи досліджують інтернет, дотримуючись посилань від однієї сторінки на іншу та індексують знайдений вміст. Під час пошуку пошукова система відображає індексований вміст. В цьому процесі сканери використовують певні алгоритми. Якщо уважно дотримуватися цих алгоритмів під час пошукової оптимізації сайту (впровадження SEO), то сайт матиме найкращі шанси з'явитися серед перших результатів, тому буде збільшуватися трафік і, відповідно, дохід (для інтернет продажів і оголошень).

Пошукові системи дають деякі рекомендації для SEO, але великі пошукові системи зберігають алгоритми та результати пошуку як комерційну таємницю. SEO поєднує в собі рекомендації пошукових систем, що є у відкритому доступі, емпіричні знання та теоретичні знання з наукових робіт або патентів.

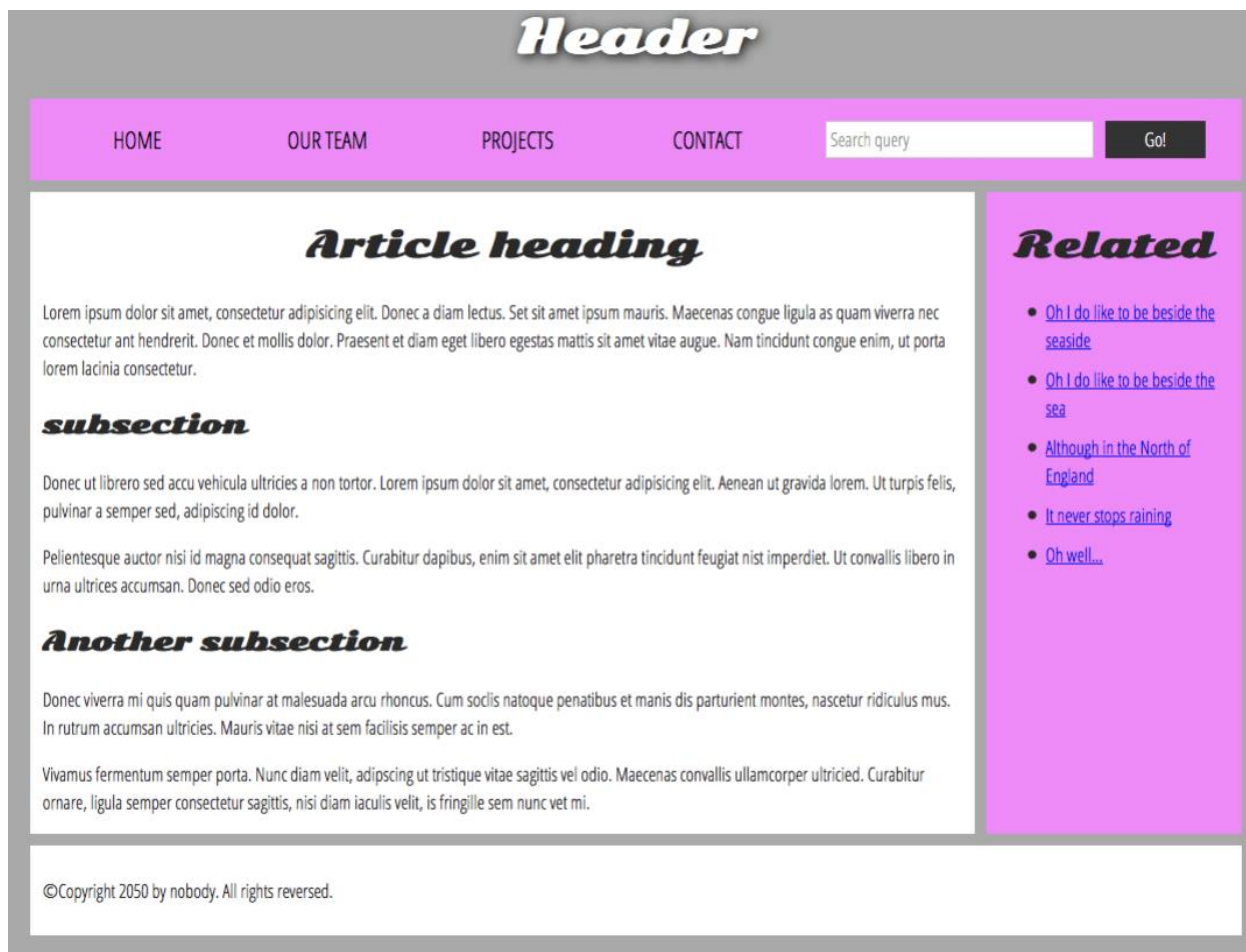
Методи SEO складаються з трьох основних класів:

Технічний. Розмітка вмісту за використання семантики HTML. Під час сканування веб-сайту, алгоритми пошуку мають знайти лише вміст, який ви хочете індексувати.

Копірайтинг. Складати контент слід за використання словникового запасу потенційних відвідувачів сайту. Використовувати текст, а також зображення потрібно так, щоб алгоритми пошуку могли зрозуміти предмет сайту.

Популярності. Значно збільшити трафік можна завдяки розташуванню посилань на даний сайт на інших популярних ресурсах.

Типовий веб-сайт може бути структурований наступним чином:



4.2. HTML для структурування вмісту

Приклад, наведений вище, не є ідеальним, але він чудово підходить для ілюстрації типової структури веб-сайту. Деякі веб-сайти мають більше колонок або складових, деякі є набагато складнішими за спецефектами, але загальна ідея залишається незмінною. Використовуючи правильний CSS, ви можете використовувати майже будь-які елементи для оформлення різних розділів та отримувати таку візуалізацію сторінок, яка вам потрібна. Проте

слід пам'ятати, що потрібно дотримуватися семантики сторінок та використовувати відповідні елементи для певного завдання.

Візуальні ефекти не відіграють найголовнішу роль під час ознайомлення зі вмістом сторінки. Можна використовувати колір і розмір шрифту, щоб привернути увагу користувачів до найбільш корисних частин контенту, таких як навігаційне меню, пов'язані посилання, але як щодо людей з вадами зору, наприклад, тих для кого концепція "рожевого" або "великого шрифту" є не дуже корисними?

У своєму HTML-кодi можна позначити розділи контенту, засновані на їх функціональності – можна використовувати елементи, які представляють розділи контенту, описані вище однозначно, і допоміжні технології на зразок екранних читців можуть розпізнавати такі елементи і допомагати із завданнями на кшталт знаходження основної навігації, або знаходження основного вмісту.

Наведемо причини, які визначають необхідність правильної побудови веб-сторінки та визначення семантичної структури.

Користувачі, які дивляться на веб-сторінку, як правило, швидко сканують, щоб знайти відповідний контент, часто просто читаючи заголовки, щоб почати з них. Зазвичай людина проводить дуже короткий час на веб-сторінці. Якщо вона не бачить нічого корисного протягом декількох секунд, вона, швидше за все, буде розчарована та покине сайт.

Пошукові системи індексування вашої сторінки розглядають вміст заголовків як важливі ключові слова для визначення пошукових рейтингів сторінки. Без заголовків сторінка працюватиме не ефективно з точки зору SEO (оптимізації пошуку).

Люди з вадами зору часто не читають веб-сторінки; замість цього вони слухають їх. Це робиться за допомогою програмного забезпечення, яке називається зчитувачем екрана. Це програмне забезпечення надає способи швидкого доступу до даного текстового контенту. Одним з методів, що використовуються такими програмами є надання схеми документу, шляхом зчитування звголовків, таким чином дозволяючи користувачам швидко знаходити потрібну інформацію. Якщо заголовків немає, користувачі будуть змушені слухати весь документ, зачитаний вголос.



Для стилізації контенту за допомогою CSS, або оживлення його за допомогою JavaScript, потрібно мати елементи, які б оформлювали відповідний вміст, для ефективної роботи з CSS/JavaScript.

Тому необхідно надавати контенту відповідну структурну розмітку.

Для структуризації контенту та семантичного позначення HTML надає виділені теги, які можна використовувати для представлення таких розділів, наприклад:

Заголовок: `<header>`.

HTML-елемент `<header>` представляє вступний контент, як правило, групу вступних або навігаційних елементів. Він може містити деякі елементи заголовку, але також логотип, форму пошуку, ім'я автора та інші елементи, які характеризують весь сайт.

HTML	CSS	OUTPUT
<pre>1 <header class="page-header"> 2 <h1>Cute Puppies Express!</h1> 3 </header> 4 5 <main> 6 <p>I love beagles so much! Like, really, a lot. They're adorable and their ears are so, so snuggly soft! </p> 7 </main> 8</pre>		
<pre>1 header.page-header { 2 background: no-repeat left/cover url(3 display: flex; 4 height: 120px; 5 min-width: 120px; 6 align-items: center; 7 color: #fff; 8 text-shadow: #000 0 0 .2em; 9 } 10 11 header.page-header > h1 { 12 font: bold calc(1em + 2 * (100vw - 12 13 fantasy; 14 margin: 2%; 15 } 16 17 main { 18 font: 1rem 'Fira Sans', sans-serif; 19 }</pre>		

Навігаційне меню: [<nav>](#).

Елемент HTML `<nav>` представляє собою розділ сторінки, метою якого є надання навігаційних посилань, або в межах поточного документа, або в інші документи. Поширеними прикладами розділів навігації є меню, таблиці вмісту та індекси.

HTML	CSS	OUTPUT
<pre>1 <nav class="crumbs"> 2 3 <li class="crumb"><a 4 href="#">Bikes 5 <li class="crumb"><a 6 href="#">BMX 7 <li class="crumb">Jump Bike 8 3000 9 10 </nav> 11 12 <h1>Jump Bike 3000</h1> 13 <p>This BMX bike is a solid step into 14 the pro world. It looks as legit as it 15 rides and is built to polish your 16 skills.</p></pre>		<p>Bikes > BMX > Jump Bike 3000</p> <hr/> <h2>Jump Bike 3000</h2> <p>This BMX bike is a solid step into the pro world. It looks as legit as it rides and is built to polish your skills.</p>

HTML	CSS	OUTPUT
<pre>1 nav { 2 border-bottom: 1px solid black; 3 } 4 5 .crumbs ol { 6 list-style-type: none; 7 padding-left: 0; 8 } 9 10 .crumb { 11 display: inline-block; 12 } 13 14 .crumb a::after { 15 display: inline-block; 16 color: #000; 17 content: '>'; 18 font-size: 80%; 19 font-weight: bold;</pre>		<p>Bikes > BMX > Jump Bike 3000</p> <hr/> <h2>Jump Bike 3000</h2> <p>This BMX bike is a solid step into the pro world. It looks as legit as it rides and is built to polish your skills.</p>

Основний вміст: [<main>](#), який складається з підрозділів за допомогою елементів: [<article>](#), [<section>](#) та [<div>](#).

Елемент `<main>` HTML представляє собою основний вміст `<body>` сторінки. Основна область змісту складається з контенту, який безпосередньо пов'язаний з центральною темою документа (центральною функціональністю застосунку) або доповнює її.

HTML	CSS	Gecko facts	OUTPUT
<pre>1 <header>Gecko facts</header> 2 3 <main> 4 <p>Geckos are a group of usually small, usually nocturnal lizards. They are found on every continent except Australia.</p> 5 6 <p>Many species of gecko have adhesive toe pads which enable them to climb walls and even windows.</p> 7 </main> 8</pre>		<p>Geckos are a group of usually small, usually nocturnal lizards. They are found on every continent except Australia.</p> <p>Many species of gecko have adhesive toe pads which enable them to climb walls and even windows.</p>	
HTML	CSS	Gecko facts	OUTPUT
<pre>1 header { 2 font: bold calc(.025 * (100vw)) 'Aria 3 } 4</pre>		<p>Geckos are a group of usually small, usually nocturnal lizards. They are found on every continent except Australia.</p> <p>Many species of gecko have adhesive toe pads which enable them to climb walls and even windows.</p>	

Елемент HTML `<article>` є автономною частиною сторінки, застосунку або сайту, яка призначена для самостійного розповсюдження (дистрибуції) або багаторазового використання. Прикладами можуть бути: пост, журнал, стаття, запис в блозі, коментар, що подається користувачем, інтерактивний віджет, або будь-який інший незалежний елемент контенту.


HTML	CSS	OUTPUT
<pre> 1 <article class="forecast"> 2 <h1>Weather forecast for Seattle</h1> 3 <article class="day-forecast"> 4 <h2>03 March 2018</h2> 5 <p>Rain.</p> 6 </article> 7 <article class="day-forecast"> 8 <h2>04 March 2018</h2> 9 <p>Periods of rain.</p> 10 </article> 11 <article class="day-forecast"> 12 <h2>05 March 2018</h2> 13 <p>Heavy rain.</p> 14 </article> 15 </article> 16 </pre>		

HTML	CSS	OUTPUT
	<pre> 1 .forecast { 2 margin: 0; 3 padding: .3rem; 4 background-color: #eee; 5 font: 1rem 'Fira Sans', sans-serif; 6 } 7 8 .forecast > h1, 9 .day-forecast { 10 margin: .5rem; 11 padding: .3rem; 12 font-size: 1.2rem; 13 } 14 15 .day-forecast { 16 background: right/contain content-box 17 url('/media/examples/rain.svg') w 18 } </pre>	

Елемент HTML `<section>` є автономним розділом документа, який не має більш точного семантичного представлення. Такі розділи завжди повинні мати заголовок (за дуже рідкими винятками).

HTML	CSS	OUTPUT
<pre>1 <h1>Choosing an Apple</h1> 2 <section> 3 <h2>Introduction</h2> 4 <p>This document provides a guide to help with the important task of choosing the correct Apple.</p> 5 </section> 6 7 <section> 8 <h2>Criteria</h2> 9 <p>There are many different criteria to be considered when choosing an Apple — size, color, firmness, sweetness, tartness...</p> 10 </section> 11</pre>		<div><h1>Choosing an Apple</h1><h2>Introduction</h2><p>This document provides a guide to help with the important task of choosing the correct Apple.</p><h2>Criteria</h2><p>There are many different criteria to be considered when choosing an Apple — size, color, firmness, sweetness, tartness...</p></div> <div>MDN Web Docs Interactive Example</div>
<pre>1 h1, h2 { 2 margin: 0; 3 } 4</pre>		<div><h1>Choosing an Apple</h1><h2>Introduction</h2><p>This document provides a guide to help with the important task of choosing the correct Apple.</p><h2>Criteria</h2><p>There are many different criteria to be considered when choosing an Apple — size, color, firmness, sweetness, tartness...</p></div>

HTML-елемент `<div>` - це загальний контейнер для потокового контенту. Він не впливає на вміст або вигляд сторінки, поки не застосовується певним чином з CSS (наприклад, стиль безпосередньо застосовується до нього, або якась модель макете на зразок Flexbox – модель оптимізації зображення - застосовується до його батьківського елемента). Використовується для групування контенту, що дозволяє легко стилізувати його за допомогою атрибутів `<class>`, `<id>`, `<lang>`.

HTML	CSS	OUTPUT
<pre> 1 <div class="warning"> 2 5 <p>Beware of the leopard</p> 6 </div> </pre>		
HTML	CSS	OUTPUT
<pre> 1 .warning { 2 border: 10px ridge #f00; 3 background-color: #ff0; 4 padding: .5rem; 5 display: flex; 6 flex-direction: column; 7 } 8 9 .warning img { 10 width: 100%; 11 } 12 13 .warning p { 14 font: small-caps bold 1.2rem sans-ser 15 text-align: center; 16 } </pre>		

Бічна панель: [`<aside>`](#), яка зазвичай розміщується всередині елемента [`<main>`](#).

Елемент `<aside>` HTML є такою частиною документа, зміст якої лише опосередковано пов'язаний з основним змістом документа. Панелі `aside` часто представлені як бічні панелі або поля для виклику.

HTML	CSS
<pre> 1 <p>Salamanders are a group of amphibians with a lizard-like appearance, including short legs and a tail in both larval and adult forms.</p> 2 3 <aside> 4 <p>The Rough-skinned Newt defends itself with a deadly neurotoxin.</p> 5 </aside> 6 7 <p>Several species of salamander inhabit the temperate rainforest of the Pacific Northwest, including the Ensatina, the Northwestern Salamander and the Rough- skinned Newt. Most salamanders are nocturnal, and hunt for insects, worms and other small creatures.</p> 8 </pre>	<p>Salamanders are a group of amphibians with a lizard-like appearance, including short legs and a tail in both larval and adult forms.</p> <p>Several species of salamander inhabit the temperate rainforest of the Pacific Northwest, including the Ensatina, the Northwestern Salamander and the Rough-skinned Newt. Most</p> <p><i>The Rough-skinned Newt defends itself with a deadly neurotoxin.</i></p>
HTML	CSS
<pre> 1 aside { 2 width: 40%; 3 padding-left: .5rem; 4 margin-left: .5rem; 5 float: right; 6 box-shadow: inset 5px 0 5px -5px #296; 7 font-style: italic; 8 color: #29627e; 9 } 10 11 aside > p { 12 margin: .5rem; 13 } 14 15 p { 16 font-family: 'Fira Sans', sans-serif; 17 } 18 </pre>	<p>Salamanders are a group of amphibians with a lizard-like appearance, including short legs and a tail in both larval and adult forms.</p> <p>Several species of salamander inhabit the temperate rainforest of the Pacific Northwest, including the Ensatina, the Northwestern Salamander and the Rough-skinned Newt. Most</p> <p><i>The Rough-skinned Newt defends itself with a deadly neurotoxin.</i></p>

Нижній колонтитул: [<footer>](#).

Елемент HTML `<footer>` - це нижній колонтитул для найближчого контенту. `<footer>` зазвичай містить інформацію про автора розділа, дані про авторські права або посилання на пов'язані документи.

HTML	CSS
<pre> 1 <article> 2 <h1>How to be a wizard</h1> 3 4 Grow a long, majestic beard. 5 Wear a tall, pointed hat. 6 Have I mentioned the beard? 7 8 <footer> 9 <p>© 2018 Gandalf</p> 10 </footer> 11 </article> 12 </pre>	

How to be a wizard

1. Grow a long, majestic beard.
2. Wear a tall, pointed hat.
3. Have I mentioned the beard?

© 2018 Gandalf

HTML	CSS
<pre> 1 article { 2 min-height: 100%; 3 display: grid; 4 grid-template-rows: auto 1fr auto; 5 } 6 7 footer { 8 display: flex; 9 justify-content: center; 10 padding: 5px; 11 background-color: #45a1ff; 12 color: #fff; 13 } 14 </pre>	

How to be a wizard

1. Grow a long, majestic beard.
2. Wear a tall, pointed hat.
3. Have I mentioned the beard?

© 2018 Gandalf

Дослідження коду прикладу

Потрібно трохи часу, щоб подивитися код і зрозуміти наведений нижче код – коментарі всередині коду також мають допомогти зрозуміти його.

Наведений код можна знайти в репозиторії GitHub.

```

<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">

    <title>My page title</title>

```

```

    <link
href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300|Sonsie+
One" rel="stylesheet">
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <!-- Here is our main header that is used across all the pages of our
website -->

    <header>
        <h1>Header</h1>
    </header>

    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Our team</a></li>
            <li><a href="#">Projects</a></li>
            <li><a href="#">Contact</a></li>
        </ul>

        <!-- A Search form is another common non-linear way to navigate
through a website. -->

        <form>
            <input type="search" name="q" placeholder="Search query">
            <input type="submit" value="Go!">
        </form>
    </nav>

    <!-- Here is our page's main content -->
    <main>

        <!-- It contains an article -->
        <article>
            <h2>Article heading</h2>

            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Donec a
diam lectus. Set sit amet ipsum mauris. Maecenas congue ligula as quam
viverra nec consectetur ant hendrerit. Donec et mollis dolor. Praesent et
diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue
enim, ut porta lorem lacinia consectetur.</p>

            <h3>Subsection</h3>

            <p>Donec ut librero sed accu vehicula ultricies a non tortor. Lorem
ipsum dolor sit amet, consectetur adipisicing elit. Aenean ut gravida lorem.
Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.</p>

            <p>Pelientesque auctor nisi id magna consequat sagittis. Curabitur
dapibus, enim sit amet elit pharetra tincidunt feugiat nist imperdiet. Ut
convallis libero in urna ultrices accumsan. Donec sed odio eros.</p>

            <h3>Another subsection</h3>

            <p>Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum
soclis natoque penatibus et manis dis parturient montes, nascetur ridiculus
mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper
ac in est.</p>

            <p>Vivamus fermentum semper porta. Nunc diam velit, adipscing ut
tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies.

```

```

Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis
velit, is fringille sem nunc vet mi.</p>
</article>

<!-- the aside content can also be nested within the main content -->
<aside>
  <h2>Related</h2>

  <ul>
    <li><a href="#">Oh I do like to be beside the seaside</a></li>
    <li><a href="#">Oh I do like to be beside the sea</a></li>
    <li><a href="#">Although in the North of England</a></li>
    <li><a href="#">It never stops raining</a></li>
    <li><a href="#">Oh well...</a></li>
  </ul>
</aside>

</main>

<!-- And here is our main footer that is used across all the pages of our
website -->

<footer>
  <p>©Copyright 2050 by nobody. All rights reversed.</p>
</footer>

</body>
</html>

```

4.3. Несемантичні елементи

Іноді зустрічаються ситуації, за яких неможливо знайти ідеальний семантичний елемент, щоб групувати деякі елементи разом або обгорнути якийсь вміст. В таких випадках слід просто згрупувати набір елементів разом, щоб вплинути на них як на один об'єкт з певними засобами CSS або JavaScript. Для таких випадків HTML надає елементи `<div>` та ``. Слід використовувати ці елементи з відповідними атрибутами класу (`<class>`, `<id>`), щоб надати якусь мітку для швидкого доступу до цих елементів.

`` - це строковий несемантичний елемент, який слід використовувати, лише якщо не можна придумати кращий семантичний текстовий елемент для оформлення вмісту, або нема потреби використовувати певні конкретні семантичні елементи. Наприклад:

```

<p>The King walked drunkenly back to his room at 01:00, the beer doing
nothing to aid
him as he staggered through the door <span class="editor-note">[Editor's
note: At this point in the
play, the lights should be down low]</span>.</p>

```

У цьому випадку, нотатка редактора повинна просто надати додаткові відомості для режисера п'єси; вона не має окремого семантичного значення.

Для користувачів з вадами зору, засобами CSS можна виокремити нотатку від основного тексту.

`<div>` - це несемантичний елемент блочного рівня, який слід використовувати лише тоді, коли не можна придумати кращий семантичний блоковий елемент, або якщо елементу не потрібно надавати конкретного семантичного значення. Наприкладом може бути віджет кошика для покупок, який користувач може відкрити у будь-який момент часу на сайті електронної комерції:

```
<div class="shopping-cart">
  <h2>Shopping cart</h2>
  <ul>
    <li>
      <p><a href=""><strong>Silver earrings</strong></a>: $99.95.</p>
      
    </li>
    <li>
      ...
    </li>
  </ul>
  <p>Total cost: $237.89</p>
</div>
```

Це не зовсім елемент `<aside>`, оскільки це не обов'язково стосується основного вмісту сторінки (ви хочете, щоб цей елемент можна було б переглянути з будь-якого місця сайту). Відсутні причини для використання елемента `<section>`, оскільки він не є частиною основного змісту сторінки. Отже, `<div>` в цьому випадку підходить найкраще.

Попередження: Елементи `<div>` настільки зручні у використанні, що їх дуже легко почати використовувати зайвою мірою. Проте, оскільки вони не несуть семантичного значення, вони просто захаращують HTML-код. Елементи `<div>` слід використовувати тільки тоді, коли немає кращого семантичного рішення і намагатися зменшити їх використання до мінімуму. В протилежному випадку буде ускладнено процес оновлення та підтримки сайту.

4.4. Перенесення рядків та горизонтальний подільник

Два елементи, потреба у використанні яких виникає під час створення кожної сторінки, це `
` та `<hr>`.

`
`: елемент розриву рядка

`
` забезпечує перенесення символів у абзаці на наступний рядок; це єдиний спосіб утворити відповідну структуру тексту, коли потрібно створити декілька фіксованих коротких рядків в межах одного абзацу, наприклад, у поштовій адресі або вірші. Наприклад:

```
<p>There once was a man named O'Dell<br>
Who loved to write HTML<br>
But his structure was bad, his semantics were sad<br>
and his markup didn't read very well.</p>
```

Без елементів `
` абзац буде відтворено в один довгий рядок (оскільки, HTML ігнорує перенесення рядків). Із застосуванням елементів коду `
` розмітка виглядає наступним чином:

There once was a man named O'Dell
Who loved to write HTML
But his structure was bad, his semantics were sad
and his markup didn't read very well.

`<hr>`: горизонтальний подільник тексту

елементи `<hr>` створюють у документі горизонтальний розподіл, який позначає тематичну зміну тексту (наприклад, зміну теми або сцени). Візуально він виглядає як горизонтальна лінія. Як приклад:

```
<p>Ron was backed into a corner by the marauding
    netherbeasts. Scared, but determined to protect his friends, he raised his
    wand and prepared to do battle, hoping that his distress call had made it
    through.</p>
<hr>
<p>Meanwhile, Harry was sitting at home, staring at his royalty statement
    and pondering when the next spin off series would come out, when an
    enchanted
    distress letter flew through his window and landed in his lap. He read it
    hazily and sighed; "better get back to work then", he mused.</p>
```

Програма покаже наступне:

Ron was backed into a corner by the marauding netherbeasts. Scared, but determined to protect his friends, he raised his wand and prepared to do battle, hoping that his distress call had made it through.

Meanwhile, Harry was sitting at home, staring at his royalty statement and pondering when the next spin off series would come out, when an enchanted distress letter flew through his window and landed in his lap. He read it hazily and sighed; "better get back to work then", he mused.

4.5. Планування простого веб-сайту

Після того, як заплановано структуру та вміст однієї веб-сторінки, наступним логічним кроком є спроба планування того, який контент має розміщуватися на всьому веб-сайті: які сторінки повинні бути, яку будову повинна мати ці сторінки, як вони мають бути пов'язані одна з одною для отримання найкращого результату на запити користувача. Така структура називається інформаційною архітектурою або архітектурою даних.

Інформаційна архітектура, яка застосовується до веб-дизайну та розробки сайтів, є практикою організації інформації / вмісту / функціональності веб-сайту із можливими сервісами, яка забезпечує такий вигляд сайту, що є найкращим з точки зору забезпечення інформаційних потреб користувача.

Планування великого та складного веб-сайту є довгим процесом, але для простого веб-сайту з декількох сторінок це може бути досить просто. Для цього слід зважати на наступні принципи побудови:

1 Має бути кілька семантичних елементів, які є спільними для більшості (якщо не всіх) сторінок – таких, як меню навігації та вміст нижнього колонтитулу. Якщо йдеться, наприклад, про бізнес-сайт, добре мати інформацію про контакти на кожній сторінці. Отже важливим є визначити той контент, який буде розташовано в колонтитулах.

Common to every page

Header: title & logo

Footer: Contact details and copyright notice

① Links to Terms + conditions

② Site language chooser

③ Accessibility policy

2 Наступним кроком має бути створення приблизного ескізу того, як буде виглядати структура кожної сторінки (за основу можна взяти виглядати простого веб-сайту наведеного на початку лекції). Слід приділити увагу наповненню, кожного блоку.



3 Наступним значним етапом є список контенту унікального для кожної сторінки сайту. Цей етап може передбачати проведення мозкового штурму всієї команди розробників

Search for Flights
Hotels / other
accommodation

Transport
Things to do

Special offers

Popular holiday packages

e.g. Winter sun
Disneyworld

Skiing

Search
results

Country-specific info

Accommodation / attraction reviews

Visa / entry requirements

Money / Currency

Languages

Insurance

Buy
holidays / other
things

4 Далі, слід впорядкувати всі ці елементи вмісту в групи, для того щоб визначитися з тим, які частини потрібно розміщувати на одних сторінках

сайту, а які на різних сторінках. Це дуже схоже на метод, який має назву сортування карт.

Сортування карт — це техніка дизайну потреб користувача, за якої розробник тестує групу експертів або користувачів для генерації дендрограми (дерева категорій) або фолксономії. Це підхід для проектування інформаційної архітектури, робочих потоків, структури меню або навігаційних шляхів у веб-сайті.

Сортування карт використовує відносно низькотехнологічний підхід. Людина, що проводить тест (аналітик юзабіліті, дизайнер потреб користувача тощо) спочатку ідентифікує ключові поняття і записує їх на індексні карти або картки нотаток. Після чого експерти / користувачі індивідуально або як група, розташовують картки, щоб надати вигляд того, як вони бачать структуру та інформаційні зв'язки на сайті.

Групи можуть бути організовані як спільні групи (фокус-групи) або як індивідуальні процедури розташування карток, які мають повторюватися.

Сортування карт зазвичай проводиться під час проектування навігаційної структури для середовища, що пропонує різноманітні контент і функції, наприклад веб-сайт.

Сортування карток використовують в наступних випадках:

1 Різноманітність предметів, які потрібно організувати, настільки велика, що жодна існуюча таксономія не приймається як організація предметів.

2 Схожість між предметами ускладнює їх чіткий поділ на категорії.

3 Члени групи експертів / користувачів, які використовують середовище, істотно відрізняються за визначенням схожості між елементами і відповідним групуванням елементів.

Search

Flights
Hotels
Other accommodation
Transport
Things to do

Country-specific info

Reviews
Visa / entry requirements
Money / currency
Languages

Specials Special offers

Popular holidays

Search results

Also include search
functionality here
to refine results

Buy things

Shopping cart
Checkout
Insurance

5. Наступним кроком є створення приблизного макету (структури) сайту – кожна сторінка на сайті може бути подана кружечком, а лінії між ними відображують переміщення користувача між сторінками під час реалізації типового робочого процесу на сайті. Головна сторінка, як правило, розташовується в центрі, а з неї користувач повинен мати можливість швидкого доступу до всіх інших сторінок сайту.

