

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА
ІННОВАЦІЙНОГО РОЗВИТКУ**

ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ
Кафедра економічної кібернетики і математичного моделювання

Веб-технології та веб-дизайн

Конспект лекцій

для здобувачів початкового рівня (короткий цикл) вищої освіти
ОПП «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки»
денної форми здобуття вищої освіти



Миколаїв
2023

Друкується за рішенням науково-методичної комісії факультету менеджменту Миколаївського національного аграрного університету від 08 березня 2023 року, протокол № 7.

Укладачі:

- О. В. Шибаніна – д-р екон. наук, професор, професор кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;
- С. І. Тищенко – канд. пед. наук, доцент, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;
- І. І. Хилько – старший викладач кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет.
- В. О. Крайній – канд. екон. наук, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;
- О. Ю. Пархоменко – канд. фіз.-мат. наук, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет.
- А. А. Чуриков – асистент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет.

Рецензенти:

- І. П. Атаманюк – д-р техн. наук, професор, професор кафедри вищої та прикладної математики, Миколаївський національний аграрний університет;
- А. В. Швед – д-р техн. наук, доцент кафедри інженерії програмного забезпечення, Чорноморський національний університет ім. Петра Могили.

Веб-технології та веб-дизайн : конспект лекцій / О. В. Шибаніна, С. І. Тищенко, І. І. Хилько, В. О. Крайній, О. Ю. Пархоменко, А. А. Чуриков.
В26 Миколаїв : МНАУ, 2023. 96 с.

Конспект лекцій призначений для вивчення теоретичних та інструментальних аспектів веб-технологій та веб-дизайну. Містить навчальні матеріали з основних тем курсу «Веб-технології та веб-дизайн», що передбачені освітньо-професійною програмою «Комп'ютерні науки» початкового рівня (короткий цикл) вищої освіти за спеціальністю 122 «Комп'ютерні науки», галузі знань 12 «Інформаційні технології. До кожної теми подаються докладні теоретичні відомості та практичні приклади їх застосування. Даний посібник буде корисним здобувачам та викладачам вищої освіти.

УДК 004.777

© Миколаївський національний аграрний університет, 2023

ЗМІСТ

| | |
|---|--------|
| Тема 1. Базові засади інтернету. Всесвітня павутина www..... | 4 9 |
| Тема 2. Етапи створення сайтів | 28 |
| Тема 3. Базові засоби розмітки сайтів | 49 |
| Тема 4. Веб-графіка і мультимедіа | 77 |
| Тема 5. Служба DNS..... | 95 |
| Рекомендована література | |

ТЕМА 1. БАЗОВІ ЗАСАДИ ІНТЕРНЕТУ. ВСЕСВІТНЯ ПАВУТИНА WWW

1. Історія заснування

У 1969 році Міністерство оборони США започаткувало розробку проекту, котрий мав на меті створення надійної системи передачі інформації на випадок війни. Метою цього проекту було створення комп'ютерної мережі з високою надійністю передачі інформації. Вихід із ладу вузла комутації або каналу передачі даних не повинен був спричиняти втрату працездатності мережі. Агентство DARPA (Defense Advanced Research Projects Agency) запропонувало розробити для цього комп'ютерну мережу (ARPANET). Розроблена мережа об'єднувала декілька навчальних закладів та дослідницьких центрів.

Перший сервер ARPANET було встановлено 1 вересня 1969 року у Каліфорнійському університеті в Лос-Анжелесі. Комп'ютер «Honeywell 516» мав 12 кілобайт оперативної пам'яті.

1 січня 1983 року мережа ARPANET перейшла з протоколу NCP на протокол TCP/IP, який досі успішно використовується для об'єднання мереж. Саме у 1983 році за мережею ARPANET закріпився термін «Інтернет».

У 1984 році була розроблена система доменних назв (англ. Domain Name System, DNS). Тоді ж у мережі ARPANET з'явився серйозний суперник — Національний науковий фонд США (NSF) заснував міжуніверситетську мережу NSFNet (англ. National Science Foundation Network). До цієї мережі за рік під'єдналось близько 10 тисяч комп'ютерів; звання «Інтернет» почало плавно переходити до NSFNet.

У 1988 році було винайдено протокол Internet Relay Chat (IRC), завдяки якому в Інтернеті стало можливим спілкування в реальному часі (чат).

У 1990 році мережа ARPANET припинила своє існування, програвши конкуренцію NSFNet. Тоді ж було зафіксовано перше підключення до Інтернету телефонною лінією. 1993 році з'явився служба WWW (word wide web) – всесвітня павутина.

У 1998 році Папа Римський Іван-Павло II заснував всесвітній день інтернету (30 вересня)

У 2004 році з'явився термін Web-2.0, а у 2007 році Web-3.0

2. Протоколи зв'язку

Обмін інформацією між серверами та клієнтами здійснюється за певними правилами, які називають протоколами. Всі дані, що циркулюють у глобальному інформаційному полі, розбито на невеликі блоки і вкладено в пакети. Кожний пакет окрім даних має заголовок, де зберігаються адреса відправника, адреса одержувача та інша інформація, необхідна для збирання пакетів у пункті призначення. Теоретично можливо, що різні пакети одного повідомлення пройдуть різними шляхами, але все одно досягнуть адресата і будуть зібрані в повне повідомлення.

Поділ даних на пакети та їх збирання у пункті призначення здійснюється під керуванням протоколу TCP (Transmission Control Protocol – протокол керування передачею), а власне передавання пакетів мережею та досягнення ними адресата забезпечує протокол IP (Internet Protocol – міжмережевий протокол). У Інтернеті використовується велика кількість протоколів, завдяки чому існує широкий спектр служб, які надаються та підтримуються за допомогою цієї глобальної мережі.

Однією зі служб є Всесвітня павутина (World Wide Web – WWW), або просто Веб. Ця розповсюджена по всьому світу інформаційна мультимедійна система, яка об'єднує в єдиному просторі інформацію різних типів. Робота у веб подібніша до віртуальної подорожі світом з індивідуванням цікавих місць. Ця служба базується на протоколі HTTP (Hyper Text Transfer Protocol – протокол передавання гіпертексту).

Інші служби базуються та називаються на основі протоколів:

- FTP (File Transfer Protocol – протокол передавання файлів). Сервери, що підтримують цей протокол, називаються FTP-серверами. Частина дискового простору таких серверів доступна через Інтернет. Крім того, до служб Інтернету належать електронна пошта, служби миттєвого передавання повідомлень, служба новин User та інші.

- SMTP (Simple Mail Transfer Protocol) - протокол передачі пошти.

- POP3 (Post Office Protocol) – пошта отримання протокол для доставлення повідомлень електронної пошти, її написання.

- Telnet (Telnational Network) – протокол для реалізації тексту інтерфейсу по мережі.

3. Адресація в Інтернеті

Для того щоб комп'ютер мав змогу передавати та приймати дані з використанням протоколу IP, він повинен мати унікальну адресу, яку називають IP адресою. На даний час використовують дві версії IP: IPv4 та IPv6.

IP версії 4 має наступний формат: xxx.xxx.xxx.xxx, де xxx — число від 0 до 255 (наприклад, 193.205.31.47). Такий формат називається точковим записом. Така адреса є 32-бітною і повинна бути унікальною в рамках однієї мережі. Таким чином, максимальна кількість комп'ютерів в рамках однієї ізольованої мережі може бути $2^{32} = 4.294.967.296$ комп'ютери.

IP версії 6 (IPv6) є новою версією протоколу, який є наступником IPv4. Новий інтернет-протокол був розроблений в 1990-х роках і, на відміну від попередньої версії, він використовує 128-розрядну систему адресації. Таким чином, адресний простір у новій версії протоколу міститиме 2128 або ж понад $3,4 \cdot 10^{38}$ адрес, що забезпечує достатнє розширення Інтернету.

IP-адреса є зручною для комп'ютерів, але людям запам'ятати її важко. Тому серверам присвоюють так звані доменні імена — набори розділених крапками послідовностей символів, наприклад mpa.u.edu.ua. Останній, найзагальніший елемент доменного імені — в нашому прикладі ua — називають доменом першого рівня, edu — доменом другого рівня і т. д. Кількість доменів у адресі не регламентовано. У багатьох країнах домен першого рівня є кодом країни: ua — Україна, fr — Франція і т. д. Домени першого рівня можуть також позначати сферу діяльності: com — комерційні компанії, gov — урядові організації, edu — навчальні заклади, org — некомерційні організації, mil — військові організації.

Проте, для відкриття будьякого документа, що зберігається на комп'ютері, необхідно вказати ім'я файлу та повний шлях до нього. Так само і для доступу до інтернет ресурсу недостатньо знати лише IP адресу або доменне ім'я комп'ютера, на якому цей ресурс

розміщено, — ви маєте вказати також папку та ім'я файлу. Крім того в Інтернеті застосовуються різні протоколи, а отже, слід вказати ще й протокол. Адресу, що містить усі зазначені елементи, називають URL (Uniform Resource Locator — єдиний вказівник на ресурс) або адресою ресурсу.

Типовий URL має такий вигляд: протокол://адреса, де протокол визначає метод доступу до ресурсу, наприклад http, ftp; адреса описує місце розташування ресурсу і включає назву сервера, шлях до документа і його ім'я. Наведемо приклади URL-адрес:

- <http://www.syhiv.com/pub/files/school.html> — адреса файлу school.html, розміщеного в каталозі pub/files на сервері www.syhiv.com. Доступ до сервера здійснюється за протоколом HTTP;

- <http://www.syhiv.com> — адреса головної сторінки сервера www.syhiv.com;

- <ftp://ftp.syhiv.com/pub/files/school.txt> — адреса файлу school.txt, розміщеного в каталозі pub/files на сервері ftp.syhiv.com. Доступ до сервера здійснюється за протоколом FTP.

4. Види хостингу

Поняття «хостинг» (hosting, host) означає сервіс з надання обчислювальних потужностей (ресурси процесора, ОЗП, дискового простору тощо) та фізичного розміщення інформаційних ресурсів (сайтів) на серверах, що знаходяться на території провайдера і постійно під'єднані до магістралей Інтернету.

Послуга хостингу, як правило, складається з надання місця, що використовується для зберігання сайтів, а також забезпечення працездатності необхідних сервісів: електронної пошти, баз даних, зберігання файлів та послуги служби DNS, які можуть надаватися як окремі послуги або міститися в комплексі.

Табл. 1.1. Класифікація типів хостингу

| | | | |
|--------------------------------------|--|---------------------------------|-------------------------|
| Фізичний хостинг Physical Hosting | Віртуальний хостинг Virtual Hosting | Хмарний хостинг Public Cloud | Спеціалізований хостинг |
|--------------------------------------|--|---------------------------------|-------------------------|

| | | | |
|--|---|---|---------------------------------|
| Оренда фізичних серверів (Dedicated hosting) | Віртуальний хостинг (Shared hosting) | Інфраструктура як сервіс (IaaS) | Адміністрування Managed Hosting |
| Розміщення серверів (Colocation) | Виділений хостинг VPS (Virtual Private Server) | Зберігання даних як сервіс (DaaS) | Хостинг персональних даних |
| | Віртуальні сервери VDS (Virtual Dedicated Server) | Платформа як сервіс (PaaS) | Хостинг DNS |
| | | Програмне забезпечення як сервіс (SaaS) | Паркування доменів |

ТЕМА 2. ЕТАПИ СТВОРЕННЯ САЙТІВ

1. Основні етапи розробки веб-сайту

Створення сайту - це трудомісткий і тривалий процес, який відбувається в кілька етапів, в міру проходження яких ідеї замовника перетворюються в реальний функціонуючий сайт.

Розробку сайту можна порівняти з будівництвом будинку, де мають послідовно бути виконані визначені етапи: від проектної документації і закладки фундаменту до внутрішньої і зовнішньої обробки приміщення.

Етапи розробки проекту, як правило, виконуються послідовно, тому, вкрай важливо дотримуватися черговості етапів і розуміти, що будь-які несподівані і неузгоджені заздалегідь зміни чи правки можуть значно вплинути на ефективність роботи.

Роботу над кожним проектом слід проводити в строгій відповідності з приведеними нижче етапами робіт з розробки сайту.

1 етап. Визначення цілей створення сайту та проведення досліджень за темою

Це найважливіший етап в створенні сайту, оскільки не можна досягти мети, якщо її немає або неправильно визначено. Від цілей буде залежати весь подальший процес створення сайту, кожен його етап. Правильно поставлена мета - це вже половина успіху.

Перш, ніж приступити до розробки, необхідно проаналізувати тему, вивчити сайти потенційних конкурентів. Надалі це допоможе в створенні власної концепції.

2 етап. Складання технічного завдання

Розробка сайту є доволі складним процесом, який вимагає детальної інформації про майбутній проект. Для досягнення повного взаєморозуміння із замовником необхідно правильно складене технічне завдання, в якому будуть відображені всі поставлені завдання. У процесі створення такого проекту проводиться поетапне узгодження виконуваних робіт, що дозволяє зробити сайт, який відповідає всім вимогам клієнта.

На цьому етапі розробник спільно з замовником визначаються з умовами щодо реалізації проекту.

- Складання технічного завдання (ТЗ) з попередніми вимогами до сайту.

- Визначення рамок бюджету, оцінювання фінансових можливостей клієнта.

- Формування команди розробників.
- Узгодження часу на реалізацію проекту.

ТЗ, як мінімум, має містити такі пункти:

- Тип сайту (лендинг, візитка, каталог, корпоративний, магазин тощо).
- Структура сайту (які сторінки повинні бути на сайті).
- Функціонал сайту (пошук, каталог, стрічка новин).
- Стиль дизайну (витриманий, молодіжний, в темних тонах), колірна гама, потреба у створенні нового бренду (логотип, слогани, фірмові кольори та шрифти).

3 етап. Технічні аспекти проектування сайту

На цьому етапі визначається архітектура сайту, файлова та логічна структура сторінок.

- **Архітектура сайту** є основною організацією клієнтської та серверної частин сайту, їх взаємозв'язок та зв'язок з кінцевим користувачем. Створення архітектури сайту є першочерговим етапом в його проектуванні.

- **Файлова структура** - це чітко оформлена система організації різних файлів. Продумана і зручна файлова структура допомагає розробнику оптимізувати роботу, а також буде зрозумілою для інших фахівців, що працюють над проектом.

- **Логічна структура сайту** - це внутрішній устрій сайту, його «кістяк», розташування сторінок, розділів, підрозділів, додаткових матеріалів. І першочерговим завданням розробників є створення стрункого порядку з хаотичного скупчення інформації.

4 етап - розробка макета дизайну сайту

Цей етап поділяється на кілька підетапів:

- Генерація ідей дизайну. Набір ідей надається замовнику у вигляді ескізів з текстовими поясненнями.

- Розробка попереднього макета дизайну головної сторінки.

- Виправлення зауважень замовника (якщо є), доробка макета до завершеного вигляду.

- Розробка внутрішніх сторінок за аналогічним алгоритмом.

У макеті повинні бути промальовані всі блоки, які будуть на сайті. Якщо до моменту розробки дизайну текстові та графічні матеріали ще не готові, можна використовувати довільний текст, але не можна залишати в макеті порожні місця.

5 етап - HTML-CSS верстка

Залежно від цілей і завдань сайту, верстка повинна задовольняти деяким вимогам. Зазвичай, ці вимоги такі:

- Кросбраузерність - сторінки повинні однаково відображатися в популярних браузерах (Mozilla Firefox, Google Chrome, Opera, Internet Explorer, Safari і т.д.).
- Адаптивність під різні розміри екрану та мобільні пристрої.
- Гнучкість верстки - можливість легко додавати/видаляти блоки інформації на сторінки.
- Швидкість обробки коду браузером.
- Валідність – відповідність до різних стандартів.
- Семантична коректність - логічне і правильне використання елементів HTML.

6 етап - програмування і встановлення на систему керування контентом (CMS)

Це чисто технічний етап, на якому реалізується весь функціонал сайту. Вимоги до цього етапу визначаються в технічному завданні.

7 етап - заповнення сайту контентом (інформацією)

На цьому етапі відбувається наповнення сайту якісним, професійним контентом. Всі матеріали сайту, будь то тексти чи графіка повинні вписуватися в загальну концепцію сайту, відповідати його цілям і задачам.

8 етап - тестування сайту і виправлення помилок

Тестування працездатності сайту може проводити як розробник, так і замовник. Найкращий варіант - це спільне тестування.

9 етап - публікація сайту в Інтернеті

На цьому етапі для сайту обирається доменна адреса, хостингова площадка, розгортання адміністративної частини або системи управління контентом, фізичне перенесення інформації. Здійснюються тестові перевірки в реальних умовах.

10 етап - просування сайту і реклама в Інтернеті

Після остаточного завершення робіт над сайтом потрібно залучати на нього відвідувачів. Для просування та популяризації сайту можна скористатися контекстною або банерною рекламою, пошуковою оптимізацією, просуванням в соціальних мережах та іншими методами.

2. Технічні аспекти створення сайту

Після визначення типу і функціоналу майбутнього сайту, розробляють **архітектуру проекту**, що враховує спільну роботу клієнтської та серверної частини, під'єднання всіх необхідних веб-додатків, бази даних, зовнішніх сервісів. Визначаються основні модулі, їх функції та взаємозв'язок.

Зазвичай, чим складніше сайт, тим складніше його архітектура і її слід ретельно проробити на початкових етапах розробки. Непродумана, погано розроблена архітектура - типова причина відмови низки сервісів на сайті, некоректного виконання функцій або відображення інформації. Це відштовхує відвідувачів, пошукові системи погано індексують інформацію, страждає репутація сайту.

Вносити корективи до архітектури сайту безпосередньо на стадіях розробки чи створення ресурсу - звичайно, можливо, але це суттєво позначиться на вартості та термінах виконання проекту. У ряді випадків доводиться все переробляти заново.

Професійна розробка архітектури сайту - те, на чому економити не прийнято, особливо, якщо мова йде не про 1-5-сторінковий ресурс, а про складний портал або Інтернет-магазин. В майбутньому прорахунки виявляються занадто дорогими.

Основні принципи, яких слід дотримуватися при проектуванні архітектури веб-сайту

- **Доступність.** Постійна працездатність та швидке надання актуальної інформації.

- **Продуктивність.** Швидке обслуговування запитів користувачів, що сприятиме довірі до ресурсу і збільшенню потоку відвідувачів.

- **Надійність.** Система повинна працювати чітко, без збоїв. Максимальна захищеність від навмисних чи випадкових пошкоджень (проникнень).

- **Здатність до масштабування.** При створенні веб-сайту необхідно враховувати його перспективи в майбутньому. Якщо цей параметр не враховувати, то при розширенні проекту можна не впоратися з великою кількістю нової інформації, що в підсумку призведе до потреби створювати новий сайт.

- **Керованість.** Потужна і зрозуміла адміністративна частина, докладна документація, технічна підтримка проекту розробниками, можливості для внесення модифікацій і оновлень.

- **Вартість.** Для повноцінного функціонування та підтримки ресурсу слід враховувати вартість програмного і апаратного забезпечення, тарифи хостингової площадки, хмарні сервіси, ліцензійні відрахування тощо.

Структура сайту - це внутрішній устрій сайту, його «кістяк», розташування сторінок, розділів, підрозділів, додаткових матеріалів. Першочерговим завданням розробників є створення стрункого порядку з хаотичного скупчення інформації. Для комфортного перебування відвідувача на сайті і для полегшення пошуку потрібної інформації, сайт повинен мати чітку і продуману структуру.

З позиції розробника, структуру сайту умовно можна поділити на два рівні:

- **На логічному рівні** сайт є сукупністю сторінок, які містять тексти, картинки, мультимедійні чи інші об'єкти. Вони об'єднані між собою єдиним дизайном, стилем і посиланнями.

- **На фізичному рівні** сайт є впорядкованим набором файлів різного типу (HTML-сторінки, зображення, програми, мультимедійні файли). Продумана і зручна файлова структура допомагає розробнику оптимізувати свою роботу, а також буде зрозумілою для інших фахівців, що працюють над проектом.

Логічна структура

Вдалий дизайн в першу чергу має орієнтуватися на відвідувача, який без особливих зусиль може знайти і використати будь яку інформацію, що міститься у сайті. При цьому повинні бути абсолютно зрозумілими схема сайту і взаємозв'язок між окремими сторінками.

Сайти з невдалою логікою не лише ускладнюють роботу відвідувачів, вони позбавляють всякого бажання на них знаходитися. Коли відвідувачі не в змозі знайти потрібну

інформацію, вони схильні припускати, що на цьому сайті її взагалі немає. Розчаровані відвідувачі навряд чи запам'ятають цей ресурс, а тим більше захочуть ще раз його відвідати.

Доцільна навігація означає, що відвідувачі точно знають, де знаходяться, де розташовані елементи сайту і як їх використовувати. Правильна структура інформації дозволяє відвідувачам без побоювання продовжувати мандрування сайтом, і бути впевненими в тому, що завжди можна без зусиль повернутися до раніше переглянутих сторінок.

Принципи логічного проектування

Логічне проектування передбачає організацію інформації на сайті, побудову його структури та навігацію по розділах. При проектуванні інформаційного наповнення сайту і засобів навігації слід дотримуватися базових принципів, які засновані на сприйнятті інформації людиною.

- **Розділення на частини.** Людина не в змозі досягнути одночасно великий об'єм інформації. Як правило, люди здатні сприймати інформацію, що містить від чотирьох до шести різних елементів. Тому, відвідувачі сайту краще орієнтуються і швидше знаходять потрібні для них матеріали, коли вони візуально поділені на групи.

- **Одноманітність.** Використання єдиної навігації по сайту з однаковим оформленням. Однакові елементи на сторінках повинні мати один розмір і знаходитися в одному і тому ж місці. Якщо сторінки, з певних причин мають відмінності, відвідувач повинен чітко розуміти, чим вони обумовлені.

- **Використання позначень.** Слід застосовувати слова, терміни та іконки, які є сталими і зрозумілими для більшості відвідувачів сайту. Не дуже вдалим є застосування піктограм, оскільки відвідувачі можуть невірно їх розтлумачити. Елементи навігації та посилання мають мати виражений зміст.

- **Доцільність.** Розділи сайту слід наповнювати інформацією, яка відповідає даному розділу чи фрагменту сторінки, недоречні елементи краще перенести в інше місце або взагалі прибрати.

Логічна організація інформації

Організація даних у вигляді впорядкованої структури повинна повідомляти відвідувача, яку інформацію він може знайти на сайті і

де її шукати. Структура сайту нагадує зміст книги: якщо його добре складено, то стає зрозуміло, що очікувати від книги до її прочитання. Інформація повинна бути організована так, щоб відвідувач знав за назвою посилання, що його чекає на наступній сторінці.

Організація інформації на сайті може бути:

Лінійною. Сторінки можна переглядати послідовно, одна за одною. З поточної сторінки можна перейти або на наступну, або на попередню (рис.2.1).

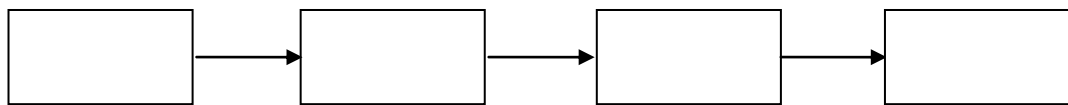


Рис. 2.1. Лінійна організація інформації

Ієрархічною. Передбачає застосування головної сторінки, на якій розміщено меню з посиланнями на розділи сайту, що розташовані на наступних сторінках (рис. 2.2). Розділи можуть містити посилання на підрозділи чи іншу детальну інформацію.

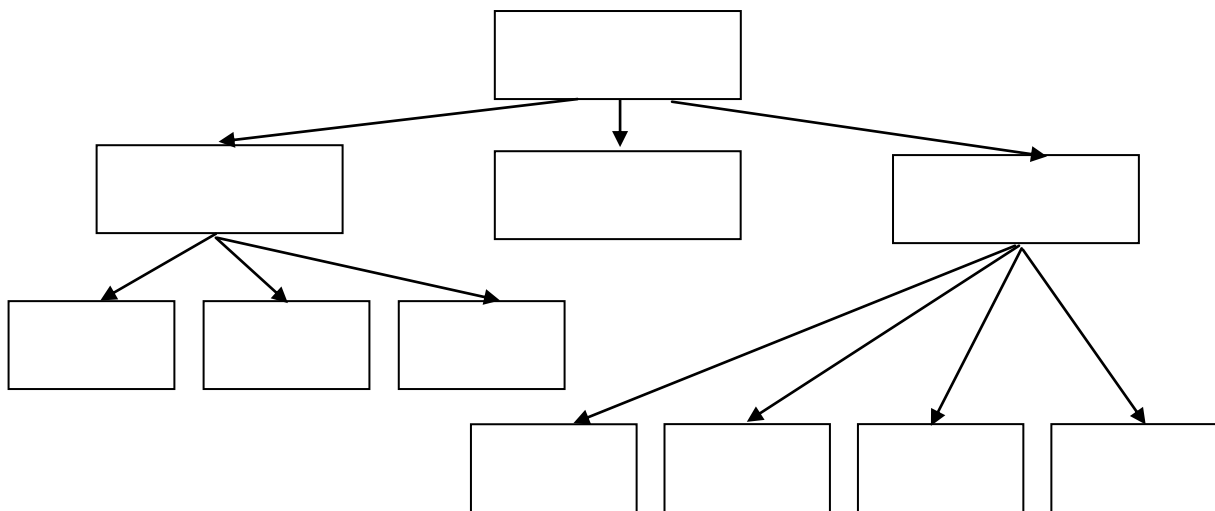


Рис. 2.2. Ієрархічна організація інформації

Блочною. Є найбільш поширеною структурою. За такої організації можна перейти з будь-якої сторінки на будь-яку іншу, що є з нею логічно взаємозв'язаною (рис. 2.3). Яскравими прикладами такої структури є Інтернет-магазини чи великі інформаційні портали.

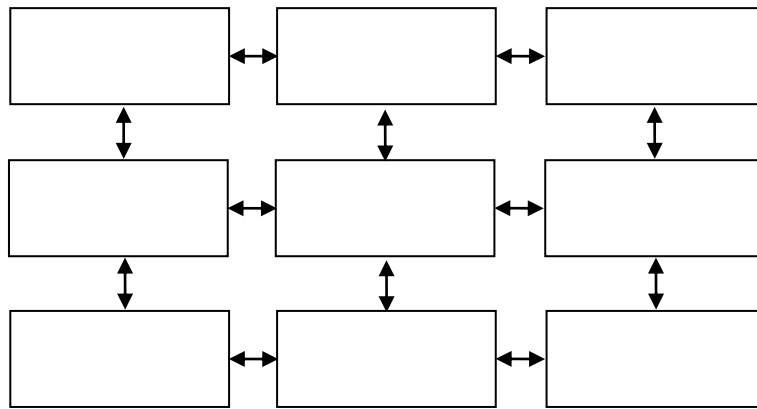


Рис. 2.3. Блочна організація інформації

Комбінована структура. Є поєднанням кількох попередніх структур. Наприклад, на сайті може бути передбачено ієрархічну структуру, яка в певному місці може містити виконання покрокових дій (лінійну структуру).

Поради з організації структури

Спроба створення структури сайту без врахування думки відвідувачів є величезною помилкою, яка може нашкодити популярності сайту. Не важливо, наскільки красивим і приємним здається сайт, вирішальним буде чи може відвідувач легко віднайти потрібну інформацію. Дизайн робиться для зручності відвідувачів, а не власників сайту.

Звичайно, можна створити логічну структуру сайту чисто інтуїтивно. Проте, однією з найбільших помилок є використання на сайтах компаній схем, що є прийнятними в цих компаніях, наприклад, розподіл товарів по брендах або повна відповідність структури сайту до структури організації. В результаті, можна створити сайт, структура якого буде очевидною для його розробників, але абсолютно незрозумілою для його відвідувачів.

Наприклад, якщо компанія займається продажем побутової техніки, то в якості критеріїв товарів краще застосовувати тип товарів, а вже потім назви фірми-виробника. Хоча назва виробника може здаватися загальновідомою для працівників компанії, вона може бути незнайомою для потенційних покупців.

Фізична структура

Під фізичною структурою проекту розуміють порядок зберігання файлів проекту в його директорії. Часто доводиться бачити, коли всі файли знаходяться разом, назви файлів зазначено

комбінацією малих та великих літер, загальними назвами типу page1, page2, цифрами, кирилицею тощо.

Це неприпустимо, по-перше, це неповага до фахівців, які будуть працювати з проектом в подальшому, по-друге, чим більше буде проект, тим більше буде ставати файлів і, зрештою можна заплутатися, що до чого відноситься і що потрібно, а що ні.

Зазвичай, в корені сайту розташовують html-документи, що відповідають кожному розділу. Сторінки підрозділів на розсуд розробника можуть міститися або в корені або рознесені до різних відповідних папок.

Всі зображення, що використовуються у сторінках, прийнято зберігати в окремих папках.

- Зображення, що використовуються в оформленні зовнішнього вигляду сторінок традиційно розміщують у папку з назвою images або img. Це сприяє стрункій систематизації файлів і є зрозумілим для розробників, що працюють над проектом. Аналогічно, для скриптів можна створити папку script, а для стилів style.

- Зображення, що відносяться до контенту певних розділів, наприклад, збірка фотографій за різні роки, то для них можна зробити окремі папки, але із змістовними назвами, наприклад, photo/2021/summer/

Дотримуючись цих правил структура проекту буде чіткою і зрозумілою (рис. 2.4).

| | |
|---|------------------------------|
| Веб-сайт - Youbook (D:\web\You-books\y... | Папка |
| css | Папка |
| bootstrap.css | 151KB Файл "CSS" |
| materialdesignicons.css | 107KB Файл "CSS" |
| mystyle.css | 13KB Файл "CSS" |
| fonts | Папка |
| materialdesignicons-webfont.eot | 286KB Файл "EOT" |
| materialdesignicons-webfont.svg | 2.11MB SVG-документ |
| materialdesignicons-webfont.ttf | 286KB Файл шрифта TrueType |
| materialdesignicons-webfont.woff | 141KB Файл "WOFF" |
| materialdesignicons-webfont.woff2 | 107KB Файл "WOFF2" |
| images | Папка |
| counter.jpg | 3KB ACDSSee Pro 3 JPEG Image |
| favicon.ico | 2KB Значок |
| favicon.png | 2KB ACDSSee Pro 3 PNG Image |
| logo.png | 7KB ACDSSee Pro 3 PNG Image |
| up.png | 2KB ACDSSee Pro 3 PNG Image |
| js | Папка |
| app.js | 8KB Файл сценария JScript |
| back-to-top.js | 4KB Файл сценария JScript |
| bootstrap.js | 58KB Файл сценария JScript |
| bootstrap.min.js | 28KB Файл сценария JScript |
| jquery-1.8.2.min.js | 92KB Файл сценария JScript |
| jquery.sticky.js | 5KB Файл сценария JScript |
| modernizr.custom.js | 9KB Файл сценария JScript |
| picture | Папка |
| index.html | 62KB Файл "HTML" |

Рис. 2.4. Файлова структура сайту

Існують також правила щодо найменування файлів проекту. Найчастіше застосовуються такі імена: головна сторінка - це `index.html`, стилі проекту `styles.css` або `*.css`, скрипти `scripts.js` або `*.js`, мінімізовані версії файлів мають префікс `.min` (`scripts.min.js`), картинки мають не розлогі назви або набори цифр, а відображають те, що на них зображено, наприклад, `button.png`, `download-icon.png`, `logo.png` і т.д.

У складі сайту мають бути присутні файли, які можна віднести до обов'язкових та службових. Зазвичай, їх розміщуються в корені сайту. Це файли, що призначені для виконання певних завдань і мають стандартну назву, а також файли, назви яких є довільними і визначаються розробниками.

Обов'язкові файли зі стандартними іменами

- `index.html` (`index.*`) — за замовченням є файлом головної сторінки, а також веб-сторінок, що розміщені в папках і повинні відкриватися при їх вказуванні в адресі. Назву і розширення

індексового файлу, можна змінити в залежності від технології створення сторінок, типу веб-сервера та його налаштувань.

- `favicon.ico` (`favicon.*`) — іконка сайту у вигляді зображення розміром 16 на 16 пікселів у форматі `.png` або `.ico`. Вона з'являється в браузері біля адреси сайту в адресному рядку, у вкладці браузера біля назви сайту, у сторінці пошукової системи біля короткого опису сайту.

Обов'язкові файли з довільними іменами

`style.css` чи `**.*css` — файл з таблицею стилів. Жоден з сучасних сайтів не обходиться без стилів, які зазвичай, розміщуються в окремому файлі. Таке розміщення надає кілька переваг — можна легко змінити вигляд елементів на всіх сторінках, за рахунок зміни параметрів лише в одному місці, файл кешується і завантажується швидше, зменшується загальний об'єм всіх документів. Файл зі стилями має розширення `.css`.

Додатково в кореневій директорії розташовують службові файли:

- `.htaccess` — конфігураційний файл веб-серверу Apache. Вказаний сервер є найбільш популярним і поширеним, тому, даний файл можна зустріти повсюдно. Бувають, звичайно, виключення і певні сервери його не сприймають.

- `robots.txt` — звичайний текстовий файл, в якому описано правила індексації сайту для пошукових машин, зокрема, забороняє роботам відвідувати певні сторінки або ділянки сайту. Наприклад, якщо певна область сайту призначена для внутрішнього користування співробітниками компанії, можна заборонити роботам її відвідувати і аналізувати, зробивши відповідні записи в `robots.txt`.

При індексуванні сайту, в першу чергу шукається саме цей файл.

- `err404.html` — файл, який буде відображено у випадку, якщо сервер не зміг знайти запрошений користувачем файл.

Хоча, назви певних службових файлів залежать від налаштувань сервера і можуть мінятися від сайту до сайту, наведений перелік є характерним для більшості з них. Для динамічних сайтів, зміст яких автоматично формується за допомогою програмних сценаріїв, файлова структура буде дещо різнитися від наведеної. Проте, і в цьому випадку, службові файли, такі як `.htaccess`, `favicon.ico`, `robots.txt`, `err404.html` є потрібними.

Поради

Для того, щоб всі посилання у сайті працювали коректно, документи відкривалися правильно і браузер не видавав помилок при зверненні до певних розділів ресурсу, при створенні його фізичної структури слід дотримуватися кількох простих правил.

- Призначати назви папок, назви і розширення всіх файлів сайту з використанням символів лише латинського алфавіту і лише в рядковому реєстрі. Такий підхід гарантує універсальність і працездатність на різних платформах.

- Не використовувати в назвах файлів пробіли, це може викликати помилки в деяких операційних системах. Для візуального розділення змістовних частин назви краще використати дефіс або символ підкреслення, наприклад, `about-us.html`, `my_cat.jpg`.

- При привласненні назв файлів прагнути, щоб вони були змістовними. Це буде корисно і під час створення сайту і згодом, коли потрібно буде внести певні зміни у сайт, а також для інших розробників, які працюють над сайтом. Важко орієнтуватися серед сторінок, що позначаються як `1.htm`, `2.htm`, `3.htm` тощо.

- На сьогоднішній день популярними стають назви - транслітерація латинськими літерами, наприклад, `rozkład-dlya-studentiv.html` або `mij_kotyk_bavytsia.png`. Це є додатковою інформацією для пошукових систем, які навчилися інтерпретувати транслітерацію в національні символи і розуміти сенс назви.

3. Інформаційне наповнення сайту

Контент (Content, вміст) — це інформаційне наповнення сайту, інформація, яку розробник складає самостійно або копіює з дотриманням відповідних законодавств. Весь контент охороняється законом про авторське право, оскільки він є продуктом інтелектуальної праці і має своїх авторів і власників.

Якісний контент - це матеріал, що поєднує ряд властивостей: унікальність, корисність, актуальність, оздоблення, інформативність. Якісний контент дозволяє сайту краще ранжуватися в пошукових системах і привертає на сайт більше відвідувачів.

Користувач, заходячи в Інтернет, має на меті певні потреби і задовольнити їх є першочерговим завданням будь-якого контенту.

- **Потреба в інформації.** Користувачеві потрібна інформація, для нього Інтернет - це величезна бібліотека, де відповіді можна знайти на будь-яке питання.

- **Потреба в розвагах.** Людина хоче відпочити і подивитися відео-ролики, пограти в онлайн-ігри, розглядати фотографії.

- **Потреба в комунікаціях.** Користувачеві хочеться з кимось поговорити, зв'язатися.

Алгоритми роботи пошукових систем стрімко розвиваються і тепер потрібно наповнювати сайт хорошим контентом. Пошукові системи враховують унікальність контенту через середній час, які користувачі проводять на сайті, читаючи статті. Враховується дата додавання матеріалу та дата останнього оновлення. Чим вона свіже - тим вище актуальність вмісту, тим швидше сайт потрапить в топ видачі.

Хороші матеріали швидше зацікавлять користувачів. Якщо написано цікаву статтю, нею стануть ділитися, збільшиться потік відвідувачів, про сайт дізнаються, складеться хороша репутація. Все це допоможе досягти основної мети – сайт стане популярним.

Типи контенту

Текст. Це статті, новини, замітки.

- **Копірайтинг.** Авторський унікальний матеріал, що ґрунтується на особистому досвіді автора, його знаннях.

- **Рерайтинг.** Дані беруться з різних джерел, обробляються, переписуються. Текст виходить унікальним, але дані, що використані для написання, запозичено ззовні.

- **Оптимізовані тексти (SEO-тексти).** Текст, що перероблений спеціально під пошукові системи і містить ключові слова. Від грамотності автора залежить, наскільки ключі будуть вдало вписані в текст, чи не будуть вони різати око.

- **Копіпаст, плагіат.** Скопійований текст з джерела без змін або з мінімальною корекцією.

Відео-контент

Це зміст сайту, що представлений в відеоформаті. Ролики, мальоване відео (дудл-відео), огляди, інструкції - кожному виду

знайдеться своє застосування. В кожного відео повинна бути мета: залучення користувачів, підвищення довіри, навчання, демонстрація продукту і так далі. І в залежності від мети вибирається вид контенту.

За призначенням відеоконтент буває:

- **Іміджевий.** Формує необхідне відношення споживача до продукції або компанії, породжує певні емоції. Це можуть бути кейси (успішні приклади послуг), відгуки споживачів.
- **Презентаційний.** Завданням роликів є візуальна демонстрація послуги, продукту, явища, користі або вигоди для споживача. Прикладом такого ролика є звичайна телереклама.
- **Вірусний.** Повинен привертати увагу, заражати ідеєю і спонукати ділитися цим роликом з іншими користувачами.
- **Навчальний.** Це відео вчить споживача конкретним діям, показує наслідки дій, розповідає, як поводити себе в різних ситуаціях, як товар або послуга допоможуть йому. Якщо товар складний в експлуатації, то відеоінструкції допоможуть з цим розібратися.
- **Соціальний.** В такому відео показується проблема та шляхи її вирішення. Подібна реклама повинна бути тонкою, ненав'язливою, максимально природною.

Самим популярним відеоконтентом є **відеоролики**. Вони поділяються на кілька видів, наприклад, можуть бути простими і постановочними, з закадровим озвученням чи музикою.

Різновид таких роликів - **дудл-відео**. Це анімований відеоролик, в якому певна особа малює в режимі реального часу. До нього можна додавати корпоративних персонажів, текст, озвучення. Динамічні ролики мимоволі утримують увагу глядача. А значить, він активно сприймає всю інформацію з дудл-відео.

На сайті можна розмістити **просту презентацію**, що створена зі слайдів. Якщо до неї додати оригінальні спецефекти, музику або голосове озвучення - за належної частки креативу вийде гідний ролик.

Ще один варіант відеоконтенту - **скрінкасти**. Це запис того, що відбувається на екрані, з додаванням пояснюючих коментарів. Так записуються навчальні відео.

Популярністю серед користувачів користуються і **GIF-анімації**. Вони короткі, але здатні передати більше інформації, ніж невеликий текст.

Залежно від цілей компанії та її аудиторії можна використовувати **мультиплікаційні ролики** або **ролики з 3D-анімацією**. Цей формат допомагає змодельовати найважчі робочі процеси і наочно показати їх споживачеві.

Відео в реальному часі - це можливість для покупця задати питання та отримати відповіді, побачити в ролику знаменитість або подивитися демонстрацію продукту, наприклад на виставці. Деякі компанії навіть проводять навчальні вебінари.

Інфографіка

Інфографіка - це графічне представлення складної інформації. Вона необхідна, коли складні дані потрібно доступно викласти для широкої аудиторії. В інфографіці використовуються різні засоби візуалізації: графіки, зображення, діаграми, таблиці, карти, схеми.

Типи інфографіки

- **Аналітична** використовується на сайтах для бізнесу і економіки. В такий інфографіці багато статистики та цифр.
- **Новинна** використовується в газетах. Завдяки неї можна коротко описати події, порівняти «до» і «після», відобразити хронологію і провести аналітику, що сталося.
- **Реконструкція** відображає хронологію подій і відтворює певний процес.
- **Презентація** є ідеальною інфографікою для бізнесу. Відмінно підійде як для реклами, так і для звітів.

Інфографіка може бути представлена в двох формах.

- В єдиній є лише один блок, де всі елементи пов'язані між собою.
- В роздільній може бути кілька блоків, які не залежать один від одного.

Категоріально інфографіки бувають:

- **Числові**, які зазвичай відображають графіки, схеми, діаграми тощо.
- **Розширені** відносяться до візуальних і часто доповнені коротким описом.
- **Покрокові** - це інструкція графічного формату.

- **Інтерактивні** використовують тільки на сайтах, оскільки на папері реалізувати задумане не вийде. Така інфографіка може змінювати показники в залежності від обраних раніше значень.

Графічні зображення

Правильно підібрані графічні файли здатні пожвавити текст, дозволяють проілюструвати слова, продемонструвати наочні приклади і привертають увагу аудиторії. Зображення мають бути належної якості і пасувати до змісту тексту.

Щоб збільшити інформативність тексту для пошукових систем варто слідувати нескладним правилам попередньої обробки.

- **Правильна назва графічного файлу.** Назва файлу має відповідати його вмісту. Так пошуковим роботам буде простіше зрозуміти, що зображено на картинці.

- **Розмір файлу.** Велика вага зображення значно уповільнює завантаження сторінки. Перед публікацією слід попередньо обробити зображення спеціальними інструментами для стиснення. Вони дозволяють зменшити розмір та зберігти при цьому якість і чіткість картинки.

- **Підписи до зображення.** Користувачі звертають увагу на текст, який супроводжує зображення, навіть коли побіжно переглядають статтю. Це така ж важлива частина структурування контенту, як заголовки, підзаголовки і списки. Звісно, що є зображення, які таких підписів не потребують

- **Атрибути Alt і Title.** Не слід нехтувати атрибутами Alt і Title, що описують зображення. Це важлива інформація для пошукових систем та пристроїв читання з екрану. Опис може бути розлогим і стосуватися саме цього графічного файлів.

- **Розміщення зображення.** Варто, щоб зображення оточував пов'язаний текстовий контент.

Принципи якісного контенту

Унікальність. Хороший контент - унікальний контент. Пошукові машини цінують авторські тексти і зображення, які не збігаються з матеріалами інших сайтів. Сторінки з унікальними статтями частіше індексуються пошуковими системами, піднімаються в топі видачі, залучають більше відвідувачів. Кращий спосіб створити унікальний контент - писати його самостійно. Це складний і довгий шлях, але витрати себе виправдовують.

Інформативність. Матеріал повинен бути корисним для відвідувачів.

Достовірність. Для цікавого матеріалу потрібні хороші джерела інформації. Публікуються лише достовірні факти, це викличе довіру читачів.

Актуальність. Контент на сайті повинен бути актуальним. Актуальність - це не просто огляд подій, які відбулися недавно. Актуальність - це те, що саме зараз читач може взяти і втілити. Посилатися можна і на факти, які мали місце кілька років тому, але все ще не втратили своєї значущості.

Грамотність. Правильна мова, письмова або усна, завжди приємна. Зараз складно знайти статті в блогах, які були б написані з помилками. Тексти мають обов'язково перевірятися на наявність орфографічних і граматичних помилок.

Структурування та ієрархія тексту. Заголовки, підзаголовки, списки. З такими текстами зручніше працювати читачеві. І пошукові боти віддають перевагу саме тому контенту, в якому є чітка структура.

Наявність доцільних посилань. Допмагають читачам орієнтуватися в матеріалах, надають можливість швидкого доступу до важливих розділів.

Траєкторія перегляду тексту на сторінці

F-патерн - це сама поширена траєкторія сканування контенту, який організований в формі блоків. "F" - тобто fast, швидко. Саме так люди з вражаючою швидкістю, буквально за кілька секунд пробігають поглядом по всій сторінці.

F-патерн отримав загальне визнання після того, як компанія Нільсен-Норман Груп провела дослідження рухів очі при читанні веб-контенту. В ході дослідження понад 200 користувачів вивчали тисячі веб-сторінок. Дослідження виявило закономірність в рухах очей користувачів при вивченні різних сайтів: загальна траєкторія сканування контенту нагадує букву "F". При цьому можна виділити три етапи сканування (рис. 2.5):

1. Спочатку користувачі читають верхню частину сторінки по горизонталі. Це верхня планка літери "F".

2. Далі, вони сканують вниз по лівому краю екрану в пошуках цікавої інформації. Натрапивши на таку інформацію, вони

починають її читати - тобто знову йдуть по горизонталі і формують середню планку літери "F".

3. І нарешті, користувачі сканують решта контенту по вертикалі вниз.

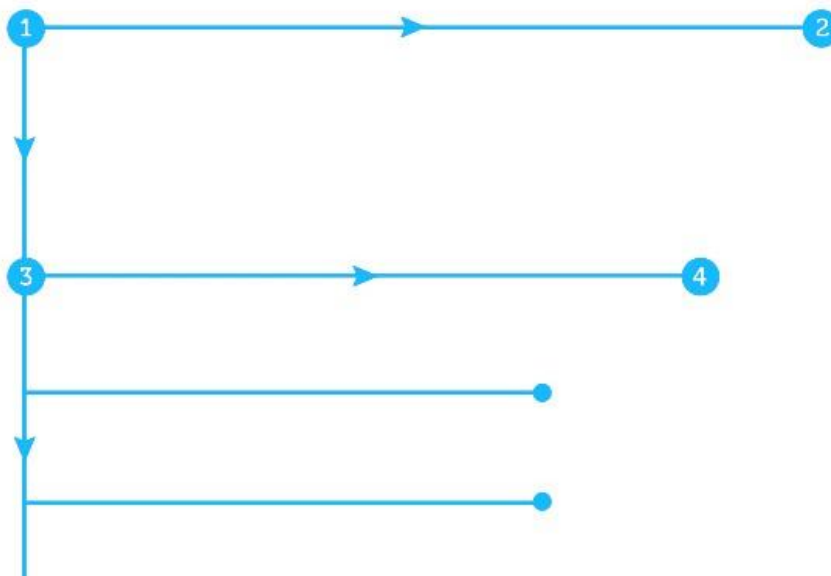


Рис. 2.5. Траєкторія перегляду тексту за F-патерном

Очевидно, що траєкторія сканування не завжди складається з трьох чітких відрізків. Коли користувач знаходить щось цікаве, він починає читати, формуючи горизонтальні лінії.

F-макет - це хороше рішення для сайтів, де багато тексту: наприклад, блогів і новинних сайтів. Якщо на сторінці багато контенту - особливо тексту - користувачам буде простіше вивчати його за природною, звичною траєкторією сканування. F-макет дозволяє дизайнеру контролювати, на що користувач зверне увагу.

Розставити контент за пріоритетністю. Перш ніж компоувати елементи на сторінці, їх слід розмістити за ступенем пріоритетності. Елементи, на які потрібно звернути увагу користувачів в першу чергу - розміщують в "гарячих точках" F-патерну, щоб домогтися потрібного ефекту.

Надання можливості швидкого ознайомлення. Перші два параграфи - найважливіші. Найважливіший контент розміщується якомога вище, щоб відразу повідомити читачеві головну ідею і призначення сайту (або сторінки). Верхній рядок користувачі, зазвичай, читають цілком по горизонталі, а значить це відмінне місце для панелі навігації.

Проектування вмісту під сканування, а не під читання. В першу чергу варто орієнтуватися на користувачів-сканерів. Додавати контент, який може їх зацікавити, на лініях F-патерну:

- Починати новий параграф з ключових слів, які привернуть увагу "сканерів".
- Люди в першу чергу звертають увагу на елементи або області з великою візуальною вагою. Збільшувати візуальну вагу важливих речей: якщо це текст, то використовувати типографіку (наприклад, виділити в тексті ключові слова), а якщо це кнопка – оформити її в яскравому кольорі.
- Один параграф - одна ідея. Частіше використовувати списки.
- Розміщення важливого контенту (наприклад, заклик до дії) зліва чи справа - там, де користувач починає і закінчує сканування. У цих точках погляд користувача на мить зупиняється - а значить, він приділить для важливої інформації дещо більше уваги.

Використання бічної панелі. Бічна панель потрібна, щоб залучати користувачів у взаємодію на більш глибокому рівні:

- Додати до бічної панелі все, що не має першорядної важливості, але заслуговує на увагу користувача. Це може бути реклама, список статей по темі, кнопки соціальних мереж і т.д.
- Перетворити бічну панель в джерело спеціалізованого контенту. Наприклад, там можна розмістити список категорій, хмару тегів, віджет з популярними постами і т.д.

Уникати довгих монотонних макетів. Основний недолік F-макету в тому, що він прагне до монотонності. Користувачам швидко набридають повторювані рядки, тому варто пожвавити область сканування якимось елементом (картинка, художня лінія, оформлена цитата тощо).

F-патерн повторює природну траєкторію руху погляду, тому можна оптимізувати під неї макет сторінки. Але не потрібно строго дотримуватися цього патерну - зрештою, це просто рекомендація, а не стандарт.

ТЕМА 3. БАЗОВІ ЗАСОБИ РОЗМІТКИ САЙТІВ

1. Розмітка документів в HTML.

HTML - не є мовою програмування, це мова розмітки, яка використовується для розміщення елементів (тексту, картинок, кнопок, таблиць тощо) на сторінці, тобто, для складання структури. Сторінка може мати складну або просту структуру, все залежить від досвіду веб-розробника.

HTML складається з ряду елементів, які використовують для того, щоб різні частини сторінки мали певний вид або спрацьовували певним способом. За призначення, вигляд та розташування елемента відповідають спеціальні оператори (теги), які представляють собою команду, що поміщена в кутові дужки. Теги, зазвичай, є скороченою назвою або аббревіатурою того завдання, що має виконати браузер наприклад:

<p> - тег, що позначає абзац (paragraph).

<tr> - тег, що позначає рядок таблиці (table row).

<h1> - тег, що позначає заголовок 1 рівня (header 1 level).

Для збільшення інформативності тегу використовуються атрибути, що вказують текстові чи числові значення.

```
<table width="50%" align="center">...</table>
```

В наведеному прикладі показано тег таблиці, яка має займати 50% від ширини вікна браузера (або блоку, в якому знаходиться) і розміщена по центру.

В мові HTML застосовують теги: одинарні, парні та коментарі.

Одинарні теги HTML

Одинарні теги складаються з одного оператора. Наприклад, тег
 - перенесення на новий рядок, <hr> - розділова лінія, - уставка зображення. Властивості для одиночного тегу вказуються всередині через атрибути, наприклад:

```

```

В наведеному прикладі в одиночному тегі прописано атрибути, які вказують на місце розташування і назву графічного файлу, ширину зображення в 200 пікселів і наявність рамки навколо зображення шириною в 1 піксел.

Певний час вимагалось наприкінці одиночного тегу вказувати слеш (
,), натепер такий стиль написання вже не є обов'язковим.

Парні теги HTML

Парних тегів є значно більше і вони вказують браузеру початок і закінчення певного структурного елемента: рядка, блоку, таблиці, заголовку

Наприклад, для того, щоб певний текст відобразився в браузері як рядок, слід застосувати теги, що позначають елемент-параграф (рис. 3.1):

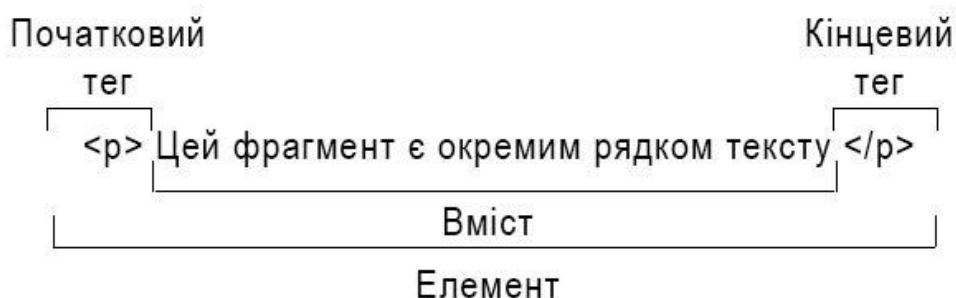


Рис. 3.1. Склад HTML-елементу

Основними частинами елемента є:

- **Початковий тег:** Він складається з назви елемента (<p>) і є ознакою початку елемента, з цього моменту тег починає впливати на той вміст, що слідує за ним.
- **Кінцевий тег:** виглядає як і початковий, але містить слеш перед назвою тега (</p>). Він вказує на закінчення елемента. Закінчення в парних тегах є обов'язковими, інакше браузер може невірно відобразити вміст.
- **Вміст:** в цьому випадку вмістом є простий текст.
- **Елемент:** початковий тег + вміст + кінцевий тег = елемент.

Всередині елементів можна вкладати інші елементи - це називається вкладеністю. Наприклад, якщо потрібно зробити акцент на певному слові і виділити його грубішим шрифтом.

<p>Цей фрагмент є окремим рядком тексту</p>

Якщо застосовується розміщення елементів всередині інших, слід обов'язково зберігати порядок вкладеності. В наступному прикладі наведено невірний запис:

`<p>Цей фрагмент є окремим рядком тексту</p>`

Елементи повинні відкриватися і закриватися правильно таким чином, щоб явно перебувати всередині або зовні один одного. Якщо вони перекриваються так, як в прикладі вище, то браузер сам приймає рішення як відображати, що може привести до непередбачуваного результату.

Теги для коментарів

Коментування в HTML необхідно для покращення читабельності коду. В коментарях, зазвичай, вказується пояснення ділянки коду, що спрощує процес редагування HTML сторінки в подальшому. Розробнику легше орієнтуватися і вносити зміни, оскільки коли коду стає дуже багато, то в ньому легко заплутатися, можна поставити зайвий тег або, навпаки, не закрити його.

Тег для коментаря починається з кутової дужки, знак оклику, два дефіси, текст коментаря, два дефіси, закінчується кутовою дужкою.

```
<!-- текст коментаря -->
```

Коментарі в HTML не відображаються на сторінці в браузері користувачеві, їх можна побачити лише в коді веб-сторінки. Для зручності слід коментувати не лише початок певного блоку коду, а й його завершення, наприклад:

```
<!-- NAVIGATION -->
```

```
<nav> ... </nav>
```

```
<!-- END NAVIGATION -->
```

Структура HTML-документа

Для сторінок будь якої складності існує стандартна структура, яка має обов'язкові елементи.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>Тестова сторінка</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Моя перша сторінка</p>
```

```
  </body>
```

```
</html>
```

- **<!DOCTYPE html>**: посилання на набір правил doctypes, яким має слідувати HTML-сторінка, також вказівка на автоматичну перевірку помилок і інші корисні речі.

- **<html>** є кореневим елементом документа. Всі інші елементи містяться всередині тегів **<html> ... </html>**. Все, що знаходиться за межами цих тегів, не сприймається браузером як код HTML і не обробляється..

- **<head>** елемент містить службову, довідкову та додаткову інформацію, яка призначена для браузера, пошукових систем, сервера і не відображається для відвідувача сторінки. В цьому елементі міститься технічна інформація про сторінку: ключові слова, короткий опис сторінки, під'єднані шрифти, стилі, скрипти, зазначається тип кодування шрифтів і багато іншого.

- **<body>** елемент містить весь контент, який бачать відвідувачі - текст, зображення, відео, анімація тощо.

Елементи, що знаходяться всередині елементу **<html>**, утворюють дерево документа, так звану об'єктну модель документа, **DOM (Document Object Model)**. Елемент **<html>** є кореневим елементом, **body** і **head** – структурними елементами.

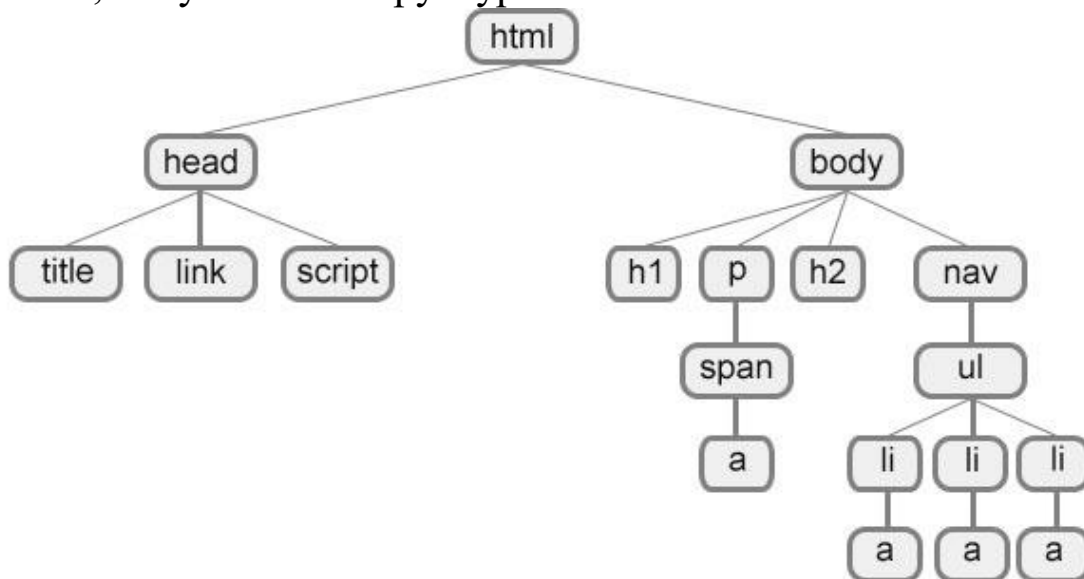


Рис. 3.2. Найпростіша структура веб-сторінки

У взаємодії елементів веб-сторінки задіяні "родинні стосунки" між елементами. Відносини між множинними вкладеними елементами підрозділяються на батьківські, дочірні та сестринські.

- **Батьківський елемент** - елемент, що пов'язаний з іншими елементами нижчого рівня, і що знаходиться на дереві вище за них. На рис. 3.2 **<html>** є батьківським лише для **<head>** і

`<body>`. Елемент `<body>` є батьківським для всіх елементів, що містяться в ньому: `<h1>`, `<p>`, ``, `<nav>` і т.д. Тег `<p>` є батьківським тільки для ``.

- **Дочірній елемент** - елемент, що розташований всередині іншого елемента. На рис.2 елементи `<h1>`, `<h2>`, `<p>` і `<nav>` є дочірніми по відношенню до `<body>`.

- **Сестринський елемент** - елемент, що має загальний батьківський елемент з даним, так звані елементи одного рівня. На рис. 3.2 `<head>` і `<body>` - елементи одного рівня, так само як і елементи `<h1>`, `<h2>` і `<p>` є між собою сестринськими.

Елемент `<head>`

Інформація всередині тегу не відображається у вікні браузера, однак, ці дані вказують браузеру, серверу чи пошуковій системі, як слід обробляти сторінку.

- **`<title>` елемент** є обов'язковим для розділу `<head>`, оскільки текст, що розміщений всередині нього, відображається у заголовку браузера. Текст заголовку повинен містити максимально повний опис вмісту сторінки.

- **`<meta>` елемент.** Елемент `<head>` може містити кілька елементів `<meta>`, які відповідно до використаних атрибутів несуть різну інформацію. За допомогою `<meta>` можна зазначити короткий опис сторінки і ключові слова для пошукових машин, автора html-документу та інші відомості.

- **`<meta charset="тип кодування">`** повідомляє браузеру кодування сторінки. Кодування HTML-сторінки потрібно вказувати для того, щоб веб-браузер міг правильно відображати текст на сторінці. Якщо браузер неправильно «вгадає» кодування, то замість тексту будуть відображатися ієрогліфи. Стандартом на сьогодні є кодування - utf-8, що містить символи зі всіх відомих природних мов, тому цей тег має вигляд `<meta charset=" utf-8">`.

- **Елемент `<style>`.** Всередині цього елемента задаються стилі, які використовуються на сторінці.

- **Елемент `<link>`.** Даний елемент визначає відношення між поточною сторінкою та іншими документами. Таких елементів на сторінці може бути кілька. Наприклад, вказати зв'язок сторінки з файлом зі стилями (окремий файл з розширенням `.css`), тоді посилання на файл зі стилями буде виглядати так `<link href="style.css">`. Обов'язковим є вказування атрибуту `rel`, де

вказується як цей документ буде розпізнано браузером (<link rel="stylesheet" href="style.css"> - документ є таблицею стилів, <link rel="icon" href="favicon.png"> - документ є фавіконкою). Додатково вказують тип документу атрибутом type (<link rel="stylesheet" type="text/css" href="style.css">, <link rel="icon" type="image/x-icon" href="favicon.png">)

- **Елемент <script>**. Дозволяє приєднувати до документа різні сценарії. Закриваючий тег є обов'язковим, при цьому текст сценарію може розташовуватися або всередині цього елемента, або в зовнішньому файлі. Якщо текст сценарію розташований в зовнішньому файлі, то він під'єднується за допомогою атрибутів елемента, наприклад: <script src="scripts.js"></script>

Елемент <body>

Всередині елемента <body> міститься інформація, яка відображається в браузері. В таблиці <https://html5book.ru/html-tags/> наведено повний перелік елементів, що підтримуються HTML5.

За призначенням теги можна умовно розподілити:

- **Контейнерні:** div, article, aside, header, footer, span, p, h1-h6, ul, ol
- **Функціональні:** a, img, table, form, input, audio, video, canvas
- **Фрейми:** iframe

У тегів можуть бути атрибути, які повідомляють браузеру, яким чином має відображатися той чи інший елемент сторінки та особливості відображення даного елемента. Значення атрибута завжди заключають в лапки " ". Назви та значення атрибутів не є чутливими до регістру, але, рекомендується набирати їх у нижньому регістрі.

<тег атрибут1="значення1" атрибут2="значення1 значення2">контент</тег>

Атрибути можна поділити на **універсальні (глобальні)** які можна використати для будь-якого HTML-елемента, та **власні**, які притаманні лише цьому тегу.

Універсальними атрибутами є

- **title** Вказує додаткову текстову інформацію про елемент, що відображається у спливаючій підказці над елементом, наприклад: <div title="Червоний блок">.

- **style** Вказує код CSS, що застосовується для оформлення даного елемента, наприклад: `<div style="width: 100%; background:red; color:#ffffff;">`.

- **class** Вказує назву класу для елемента, сам клас описано в таблиці стилів, наприклад: `<div class="red_box">`. Атрибут *class* застосовний до всіх HTML-елементів. Кожному конкретному елементу можна привласнити **кілька значень class**. Множинні значення *class* записуються через пробіл, наприклад, `<div class = "nav top">`. Назви класу повинні складатися з букв, цифр, дефісів і нижніх підкреслень і починатися тільки з літери.

- **id** Вказує унікальний ідентифікатор елемента, наприклад: `<div id="first_item">`. Атрибут *id* застосовний до всіх HTML-елементів. Кожному конкретному елементу можна привласнити **лише одне значення id**. Назви *id* повинні складатися тільки з букв, цифр, дефісів і нижніх підкреслень і повинні починатися тільки з літер.

Умовно елементи можна поділити на групи

- **Блокові елементи.** Характеризуються тим, що починаються з нового рядка, займають всю доступну ширину, висота елемента визначається його вмістом.

- **Рядкові елементи.** Є безпосередньою частиною іншого елемента, наприклад, текстового абзацу. В основному використовуються для зміни вигляду тексту, його логічного виділення чи акцентування.

Блокові елементи

Загальні характеристики блокових елементів

- На початку і в кінці блокових елементів, автоматично ставиться перенесення рядка.

- За замовченням блокові елементи займають всю ширину батьківського елемента.

- В блокових елементах можна керувати шириною і висотою: `width, height`.

- В блокових елементах можна керувати зовнішніми і внутрішніми відступами.

- Об'єктами, що знаходяться всередині блокового елемента, можна керувати за допомогою горизонтального та вертикального вирівнювання.

- Всередині блокових елементів можна розміщувати інші блокові елементи, а також рядкові елементи.

Типові блокові елементи

- **<div>** Універсальний блоковий елемент.
- **<article>**, **<aside>**, **<header>**, **<footer>** Семантичні блоки для позначення структурних ділянок сторінки.

- **<p>**, **<h1>**-**<h6>**, **<blockquote>** Елементи для HTML тексту.

- **<hr>** Горизонтальна лінія.

- ****, **** Встановлює нумерований (маркірований) список.

- **<table>** Створює таблицю.

- **<form>** Встановлює форму на веб-сторінці.

Рядкові елементи

Загальні характеристики рядкових елементів

- На відміну від блокових, рядкові елементи слідують один за одним і НЕ переносяться на новий рядок.

- В рядкових елементах не можна керувати шириною і висотою: width, height.

- В рядкових елементах можна встановлювати лише внутрішні відступи (padding) та зовнішні бічні відступи (margin).

- Рядкові елементи НЕ займають всю ширину батьківського елемента.

- Оскільки рядкові елементи НЕ займають всю ширину батьківського елемента, то елементами, що знаходяться всередині рядкових елементів, не можна керувати за допомогою горизонтального чи вертикального вирівнювання.

- Всередині рядкових елементів можна розміщувати інші рядкові елементи, якщо розмістити там блоковий елемент, то він розірве логічний потік і розміститься з нового рядка.

Типові рядкові елементи

- **<a>** Є одним з важливих елементів HTML і призначений для створення посилань.

- **** Універсальний рядковий елемент.

**
** Розірвання рядка.

**** Уставляння зображення.

HTML текст

Елементи для форматування тексту несуть змістовне навантаження і, зазвичай, задають для тексту структурне та стильове оформлення, наприклад, виділяють текст грубішим шрифтом або відображають його іншим шрифтом.

Грамотно відформатований текст надає для пошукових систем відомості, які фрагменти тексту мають важливий зміст, за якими з них слід ранжувати веб-сторінку в пошуковій видачі.

Елемент абзацу <p>

Розділяє текст на окремі абзаци, відокремлюючи їх збільшеною відстанню. Браузер автоматично додає верхній і нижній відступ.

Елементи заголовків <h1>-<h6>

Заголовки є важливим елементом веб-сторінки, вони допомагають впорядкувати текст, сформувати його структуру. У специфікації HTML існує шість рівнів заголовків, завдяки яким можна легко виділяти теми і підтеми.

Заголовки є блоковими елементами, завжди починаються з нового рядка. Браузер автоматично додає перед заголовком і після нього збільшені відступи.

Текст в заголовках впливає на індексацію сайту пошуковими системами, оскільки багато роботів звертають увагу саме на вміст заголовків сайту, тому, краще завжди використовувати ці елементи, міняючи зовнішній вигляд і розмір за допомогою стилів.

При використанні заголовків необхідно враховувати їх ієрархію, тобто за <h1> повинен слідувати <h2> і т.д. Також, не рекомендується вкладення інших елементів в <h1> ... <h6>.

Елементи для форматування тексту

- **** задає грубіше накреслення шрифту, вказуючи браузеру на важливість тексту. Є сучасною альтернативою для попереднього елемента .

- **** відображає текст нахиленим шрифтом (курсивом). Є сучасною альтернативою для елемента <i>, який на сьогоднішній день охоче застосовують для розміщення іконок. Для нахилу тексту можна використати елемент <cite>.

- **<q>** додавання зовнішніх лапок до фрагменту.

- `<code>` зміна шрифту на більш технічний шрифт.
- `` перекреслений текст.
- `<ins>` підкреслений текст.
- `<sub>` використовується для створення нижніх індексів. Зсуває текст нижче рівня рядка, зменшуючи його розмір.
- `<sup>` використовується для створення ступенів. Зсуває текст вище рівня рядка, зменшуючи його розмір.
- `<pre>` дозволяє вивести текст на екран, зберігши початкове форматування. Пробіли і переноси рядків при цьому не видаляються.
- `<blockquote>` Використовується для оформлення фрагменту особливим блоком, наприклад, цитати, виділяючи його відступами і переносами рядків.
- `
` Переносить текст на наступний рядок, створюючи розрив рядка.
- `<hr>` Використовується для поділу контенту на веб-сторінці. Відображається у вигляді горизонтальної лінії.

Гіперпосилання

HTML посилання складають основу Веб, вони пов'язують між собою різні веб-сторінки.

Посилання, або гіперпосилання, створюються за допомогою тегу `<a>`. Посилання складається з двох частин: покажчика посилання та адресної частини посилання, наприклад:

```
<p>Замовити книжки у <a href="http://shop.ua">магазині</a></p>
```

Покажчик посилання представляє фрагмент тексту або зображення, який візуально виділяється в документі (за замовчуванням, синім кольором і підкресленням).

Адресну частину посилання для користувача не видно, вона представляє URL-адресу об'єкта, до якого необхідно перейти.

Уставляння гіпертекстових посилань в HTML-документ *Uniform Resource Locator, URL*

Єдиний покажчик ресурсів (URL) визначає місцезнаходження ресурсу. В загальному вигляді має наступний формат:

метод://IP адреса сервера|доменна адреса сайту/шлях#якір.

Адреси описують, де знаходиться ресурс, наприклад, www.site.ua. Якщо адреси не вказано, то посилання вважається

локальним, тобто, воно відноситься до тієї ж машини, на якій знаходиться html-документ, що містить посилання.

Далі вказується шлях (частковий або повний), за яким здійснюється перехід.

Під якорем розуміють посилання на місце всередині поточного html-документа.

Сам текст URL не відображається браузером, він використовується для виконання запропонованих дій при активації посилання (зазвичай, клацанням миші).

Найбільш поширені методи доступу: http (https), mailto, tel, viber.

- **Метод http (https)** надає доступ до веб-сторінки за протоколом HTTP, наприклад:
НУ "Львівська політехніка"

- **Метод mailto** запускає сеанс поштового зв'язку із зазначеним адресатом і хостом, наприклад:
Iryna.Y.Yurchak@lpnu.ua
>

- **Метод tel** забезпечує з'єднання з програмою для здійснення телефонних дзвінків. Такий метод спрацьовує в мобільних пристроях і користувач може відразу позвонити за вказаним номером, наприклад: (067)947-61-21

- **Метод viber** здійснює з'єднання з месенжером Viber, наприклад:

- **Абсолютний URL**

Абсолютний шлях містить всю інформацію, що необхідна браузеру для знаходження файлу: назву протоколу, доменну або IP-адресу комп'ютера (хосту), папку (шлях до файлу) та ім'я файлу. Наприклад,

`https://www.site.ua/folder/file.html`

Якщо файл знаходиться в кореневій папці, то він вказується відразу:

`https://www.site.ru/file.html`

У разі відсутності назви файлу буде завантажуватися веб-сторінка, яка задана за замовченням у налаштуваннях веб-сервера (зазвичай, це index.*):

<http://www.site.ua/>

<http://www.site.ua/folder/>

Наявність закриваючого слешу / означає, що звернення йде до папки, а якщо його немає - то безпосередньо до файлу.

Відносний URL

Відносна URL-адреса описує шлях до зазначеного документа відносно поточного, вона визначається з врахуванням місця розташування веб-сторінки, на якій знаходиться посилання. Відносні посилання використовуються при створенні посилань на інші документи на одному сайті. У такому випадку не вказується протокол або домен, а тільки сам шлях до файлу.

1. Наприклад, у разі переходу зі сторінки <http://site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/folder2/file2.html>, тобто потрібна веб-сторінка знаходиться в тій же папці, що і веб-сторінка, що містить посилання, то посилання буде мати наступний запис:

`Текст посилання`

2. Якщо зі сторінки <http://www.site.ua/folder1/folder2/file1.html> потрібно перейти на сторінку <http://www.site.ru/folder1/folder2/folder3/file2.html>, то посилання буде таким:

`Текст посилання`

3. При переході зі сторінки <http://www.site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде мати наступний вигляд:

`Текст посилання`

3. У разі переходу з <http://www.site.ua/folder1/folder2/folder3/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде таким:

`Текст посилання`

Зовнішні посилання

Посилання на веб-сторінки інших сайтів є зовнішніми, для них завжди вказують Абсолютний URL, наприклад:

```
<a href="https://english-films.co/">Фільми англійською мовою</a>
```

Посилання на розділи поточної сторінки

Внутрішні посилання посилаються на різні частини поточної сторінки, що сприяє швидкому переміщенню, коли на сторінці багато тексту.

Внутрішні посилання також вставляються за допомогою тегу `<a>` з різницею в тому, що атрибут `href` містить назву покажчика (ідентифікатор), а не URL-адресу. Перед назвою покажчика ставиться знак `#`.

Наприклад, наступний запис описує код зі швидкими переходами на відповідні розділи.

```
<h1>Пори року</h1>
<h2>Зміст</h2>
  <a href="#p1">Літо</a> <!--Задаємо якір через id елемента-->
<
  <a href="#p2">Осінь</a>
  <a href="#p3">Зима</a>
  <a href="#p4">Весна</a>
<p id="p1"> ... </p> <!--Додаємо відповідний id елементу-->
<p id="p2"> ... </p>
<p id="p3"> ... </p>
<p id="p4"> ... </p>
```

Якщо потрібно зробити посилання з однієї сторінки сайту на певний розділ іншої сторінки сайту, то необхідно задати `id` для цього розділу сторінки, а потім додати його до абсолютної або відносної адреси:

```

<a href="http://html5.ua/page.html#picture1">Вигляд
картинки</a>
Або
<a href="page.html#picture1" target="_blank"> Вигляд
картинки</a>
```


Посилання на сторінки одного сайту

При створенні посилань на сторінки власного сайту задаються відносні посилання, які звертаються до певної сторінці на одному сервері.

Коли браузер не знаходить у посиланні протокол `https://`, він виконує пошук зазначеного документа на тому ж сервері. Відносні посилання описують шлях до потрібного файлу відносно розташування поточного документа.

У випадку, якщо сторінки знаходяться в одному каталозі, то замість запису

```
<a href="http://html5book.ua/styling.html">Оформлення  
гіпертекстових посилань</a>
```

досить буде вказати ім'я файлу з розширенням (якщо воно задане в адресі сторінки)

```
<a href="styling.html">Оформлення гіпертекстових  
посилань</a>.
```

Якщо документи знаходяться в різних каталогах, і кінцевий файл знаходиться у вкладеному каталозі відносно поточного, необхідно вказати назву каталогу і назву сторінки через слеш, наприклад,

```
<a href="lessons/lesson1.html">Урок 1</a>
```

При створенні посилання на файли з батьківського каталогу, шлях до файлу повинен починатися з `../`, що дослівно означає: повернутися на рівень вище і слідувати за зазначеним шляхом, аналогічно з кнопкою "Назад".

Можна задавати шлях до документів відносно кореневого каталогу. Звернення до кореневого каталогу здійснюється через символ `/`. Далі вказуються каталоги, в які браузер повинен перейти, щоб дістатися до потрібного документа.

```
<a href="/lessons/lesson1.html">Урок 1</a>
```

Елементи, до яких застосовують посилання

Засобами HTML можна створювати посилання за допомогою тексту, зображень та блокових елементів, тобто, можна об'єднувати в одному посиланні текст, зображення та блоки:

```
<a href="http://summer.ua">  
  <h1>ЛІТО</h1>  
  
```

```
<div>
  <ul>
    <li>Пропозиції по відпочинку</li>
    <li>Акції та знижки</li>
  </ul>
</div>
</a>
```

Атрибути посилань

Окрім глобальних атрибутів, елемент `<a>` підтримує власні атрибути.

- **href** - значенням атрибута є URL-адреса документа, на яку вказує посилання.

- **nofollow** - забороняє роботам пошукової системи переходити за посиланнями на даній сторінці або за конкретними посиланнями.

- **target** - вказує на те, в якому вікні повинна відкриватися сторінка або документ, до якого веде посилання. Приймає наступні значення:

- _self** - сторінка завантажується до поточного вікна (вкладки).
Значення за замовченням;

- _blank** - сторінка відкривається в новому вікні (вкладці) браузера;

- _parent** - сторінка завантажується у фрейм-батько;

- _top** - сторінка завантажується в повне вікно браузера.

2. Основи CSS

CSS (Cascading Style Sheets), або каскадні таблиці стилів, описують правила відображення та розміщення окремого елемента веб-сторінки. Правила створення стилю складається з двох основних частин: селектора і блоку оголошення.

Селектор повідомляє браузеру, який саме елемент форматувати, в блоці оголошення перелічено властивості форматкування та їх значення.



Рис. 3.3. Структура оголошення стилю елемента в CSS

Додавання CSS до веб-сторінки

Вбудовані таблиці стилів

Вбудовані стилі знаходяться між тегами `<style> ... </style>`, що вставляються всередину елемента `<head>`. Вбудовані стилі діють тільки на сторінці, на якій вони містяться. 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>
```

Внутрішні стилі елементів

Внутрішні стилі елементів не використовують селектори, опис стилю відбувається безпосередньо через атрибут `style` в початковому тезі елемента:

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

Зовнішні таблиці стилів

Зовнішня таблиця стилів представляє текстовий файл з розширенням `.css`, в якому знаходиться весь набір стилів CSS. Задані в файлі стилі будуть працювати для всіх сторінок веб-сайту. Під'єднати зовнішній файл зі стилями можна в два способи:

Прикріплення до веб-сторінки за допомогою тега `<link>`, вкладеного в тег `<head>`:

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

Прикріплення до веб-сторінки за допомогою правила @import

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

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

Селектори

За допомогою селекторів створюються CSS-правила для форматування елементів сторінки. Як селектори можна використовувати елементи, класи, ідентифікатори, а також псевдокласи і псевдоелементи.

Універсальний селектор. Універсальний селектор позначає правила, що стосуються всіх елементів, наприклад, наступний запис обнулить відступи для всіх елементів веб-сайту:

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

Селектор Елементу. Селектори елементів використовуються для визначення стилів для всіх даних елементів 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, які можна застосувати для різних елементів, розміщених в різних частинах або на різних сторінках сайту.

Код HTML Код CSS

```
<h1 class="headline">Інструкція користування персональним комп'ютером</h1>
```

```
<p class="headline">Примітка. Це важливо для роботи</p>
```

```
headline {
text-transform: uppercase;
color: lightblue;
}
```

Селектор Ідентифікатора. Селектори ідентифікатора використовуються для привласнення стилю одному конкретному

елементу. Ідентифікатор `id` належить унікальному елементу, тому його можна використовувати в документі лише один раз.

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

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

```
ul li {text-transform: uppercase;}
```

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 відносяться до класу з назвою `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 елемента `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 Атрибуту. Селектори атрибутів дозволяють формувати елементи на основі вибірки будь-яких атрибутів, що містяться в них або значень атрибутів, наприклад:

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

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

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

Псевдокласи

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 літера або перший рядок одного абзацу. Псевдокласи добре проілюстровані різними станами посилання, наприклад:

a: link – описує стиль невідвідуваного посилання.

a: visited - описує стиль вже відвіданого посилання.

a: hover - описує вигляд елемента, над яким проходить вказівник мишки.

a: focus - описує вигляд елемента, над яким знаходиться (сфокусований) вказівник.

a: active - описує вигляд елемента, який активовано користувачем.

Структурні псевдокласи. Структурні псевдокласи форматують дочірні елементи відповідно до зазначеного параметра в дужках, наприклад:

: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. Вказують 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 дочірній елемент 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 не є елементами сторінки, їх використовують для додавання вмісту, який генерується за допомогою властивості content, і для зміни зовнішнього вигляду частини елемента:

:before – додає певний вміст перед елементом.

:after - додає певний вміст після елемента.

Комбінації 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.

Угруповання селекторів

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

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

Принцип каскадування і специфічність правила

Каскадування представляє процес застосування різних правил до одного і того ж елемента. 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 елемента і псевдоелемента додається 0, 0, 0, 1;
- для стилю, який доданого безпосередньо до елемента - 1, 0, 0, 0;

- універсальний селектор не має специфічності.

```

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 */
#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 елемента 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`. Таке оголошення буде мати пріоритет над всіма іншими правилами.

Більше інформації по правилах CSS можна дізнатися з специфікації мови та довідників.

ТЕМА 4. ВЕБ-ГРАФІКА І МУЛЬТИМЕДІА

1. Графічні файли для Веб

У підготовці зображень для веб-сторінок варто бути обізнаним в популярних графічних форматах та їх властивостях.

На щастя, сьогодні немає такого різноманіття форматів, як на початку 90-х, коли кожна компанія-виробник графічних редакторів створювала свої файлові формати, проте кожен, з усталених на сьогодні форматів, пройшов природний відбір, довів свою життєздатність і користь. Вони мають характерні особливості і можливості, що роблять їх незамінними в роботі.

Комп'ютерні графічні файли можна поділити на дві великі гілки: растрову і векторну.

- **Векторні файли** представляють математичний опис об'єктів відносно точки відліку координат. Наприклад, для того, щоб відобразити пряму потрібно вказати координати двох точок, які об'єднуються за коротшим шляхом, для дуги задається радіус тощо. Таким чином, векторне зображення є набором геометричних примітивів. Більшість векторних форматів можуть містити втілені у файл растрові об'єкти. Складність при переведенні чи перенесенні даних з одного векторного формату до іншого полягає у використанні в програмах різних алгоритмів, різних математичних формул для побудови векторних примітивів та опису растрових об'єктів.

- **Растровий файл** влаштовано простіше. Він представляє прямокутну матрицю (*bitmap*), що поділена на піксели. Растрові формати різняться між собою здатністю містити додаткову інформацію: різні колірні моделі, вектори, Альфа-канали, прошарки різних типів, інтерліньяж (черезрядкове підвантаження), анімацію, можливості стиснення тощо.

Растрові веб-формати

GIF (CompuServe Graphics Interchange Format)

<https://www.w3.org/Graphics/GIF/spec-gif89a.txt>

Апаратно незалежний формат GIF розроблено в 1987 році (GIF87a) фірмою CompuServe для передачі растрових зображень по мережах. У 1989-му формат модифіковано

(GIF89a), до нього додано підтримку прозорості та анімації.

GIF використовує LZW-компресію, що дозволяє добре стискати файли, в яких багато ділянок з однорідним заповненням (логотипи, написи, схеми).

Метод стиснення LZW (Lempel-Ziv-Welch) розроблено в 1978 році ізраїльтянами Лемпелом і Зівом, і допрацьовано пізніше в США. Принцип стискання полягає у пошуку однакових послідовностей - фраз у всьому файлі. Виявлені послідовності зберігаються в таблиці, їм привласнюються короткі маркери - ключі. Так, якщо в зображенні є набори з рожевого, оранжевого і зеленого пікселів, що повторюються 50 разів, LZW виявляє цей набір, привласнює йому окреме число (наприклад, 7) і зберігає ці дані 50 разів у вигляді числа 7.

Метод LZW ефективно діє на ділянках однорідних, вільних від шуму кольорів, добре стискає довільні графічні дані, але процес кодування і розпаковування відбувається відносно повільно.

GIF дозволяє записувати зображення «через рядок» (Interlaced), завдяки чому, маючи лише частину файлу, можна побачити зображення цілком, але з меншою роздільністю. Це досягається за рахунок завантаження, спочатку 1, 5, 10 і далі рядків, за другим проходом підвантажуються 2, 6, 11 рядки, і згодом зображення набуває початкового вигляду. Черезрядковий запис дещо збільшує розмір файлу, але надає більшої зручності для користувачів.

У GIF можна застосовувати прозорі ділянки, вони лишаються прозорими в браузерях та інших програмах і через них просвічується фоновий колір. Прозорість забезпечується за рахунок додаткового Alpha-каналу, що зберігається разом з файлом.

Файл GIF спроможні містити кілька растрових картинок, які браузері підвантажують одну за іншою із вказаною у файлі частотою. Так досягається ілюзія руху (GIF-анімація).

Особливості

- Кількість кольорів в зображенні може бути від 2 до 256, але це можуть бути будь-які кольори з 24-бітової палітри.
- Файл у форматі GIF може містити лише 100% прозорі ділянки. Якщо використовується відмінний від білого кольору фон, він буде просвічуватися крізь прозорі ділянки в зображенні.

- GIF підтримує покадрову зміну зображень, що робить формат популярним для створення банерів і простої анімації.
- Використовує вільний від втрат метод стиснення.

Область застосування

Текст, логотипи, ілюстрації з чіткими краями, анімовані малюнки, зображення з прозорими ділянками, банери.

JPEG (Joint Photographic Experts Group)

<https://jpeg.org/>

JPEG це є назва формату та алгоритму стиснення, який засновано не на пошуку однакових елементів, як у LZW, а на різниці між пікселями.

Кодування даних відбувається в кілька етапів. Спочатку графічні дані конвертуються в колірний простір типу LAB, відкидається половина або три чверті інформації про колір (в залежності від реалізації алгоритму).

Далі аналізуються блоки 8x8 пікселів. Для кожного блоку формується набір чисел. Перші кілька чисел представляють колір блоку в цілому, в той час, як подальші числа відображають тонкі деталі.

На наступному етапі, залежно від обраного рівня якості, відкидається певна частина чисел, що представляють тонкі деталі. На останньому етапі використовується кодування за методом Хафмана для ефективного стиснення кінцевих даних. Відновлення даних відбувається в зворотному порядку.

Метод стиснення Хафмана (Huffman) розроблено в 1952 році і використовується як складова частина в ряді інших схем стиснення, в тому числі і у LZW. В методі Хафмана аналізується набір символів для визначення частоти кожного символу. Для символів, що найчастіше зустрічаються, використовується позначення у вигляді мінімальної можливої кількості бітів. Наприклад, найчастіше в англійських текстах зустрічається буква "e". Використовуючи кодування Хафмана можна представити літеру "e" лише двома бітами (1 і 0), замість вісьмох бітів, необхідних для представлення букви "e" в кодуванні ASCII.

Таким чином, чим вище рівень компресії, тим більше даних відкидається, тим нижчою є якість. Використовуючи JPEG можна отримати файл в 2-500 разів менше, ніж BMP.



Формат JPEG є апаратно незалежним, повністю підтримується на PC і Macintosh. JPEG відтворює спектр кольорів TrueColor (2^{24}).

JPEG краще стискає растрові картинки фотографічної якості, ніж логотипи або схеми - в них більше півтонових переходів, в той час серед однотонних заливок з'являються небажані переходи.

Краще і з меншими втратами стискаються великі зображення для Веб з високою роздільністю (200-300 dpi і більше), ніж з низькою (72-150 dpi), оскільки в кожному квадраті 8x8 пікселів переходи виходять м'якшими, за рахунок того, що таких квадратів є більше.

Даний формат називають стисненням з втратами, оскільки алгоритм JPEG вибірково відкидає дані. Не бажано зберігати у JPEG-форматі будь-які зображення, де важливими є тонкі нюанси кольорів, оскільки під час стиснення відбувається відкидання колірної інформації. У JPEG слід зберігати лише кінцевий варіант роботи, оскільки кожне повторне збереження призводить до нових втрат (відкидання) даних і початкове зображення може бути вкрай зіпсованим.

Формат JPEG не підтримує прозорість і при збереженні зображення з прозорими ділянками, вони зафарбовуються в певний колір.

Особливості

- В зображенні може бути понад 16 мільйонів кольорів, що цілком достатньо для збереження фотографічних зображень.
- Основною характеристикою формату є якість, яка визначає кінцевий розмір файлу. Слід пам'ятати, що формат застосовує стиснення з втратами. Чим вище стиснення, тим менше якість і навпаки.
- Підтримка технології прогресивного JPEG. Спочатку у вікні перегляду з'являється версія зображення з низькою роздільністю, яке при повному завантаженні поступово набуває початкового вигляду.

Область застосування

Використовується переважно для фотографій. Не є доцільним для зображень з прозорими ділянками, великими одноколірними ділянками.

PNG (Portable Network Graphics)

<http://www.libpng.org/pub/png/>

PNG – Інтернет формат, який долає обмеження GIF. Використовує стиснення без втрат Deflate, подібне до LZW. Стиснуті індексовані файли PNG, зазвичай, є меншими за аналогічні GIF.

Глибина кольору може бути любою, до 48 біт. Використовується двохвимірний interlacing (не лише рядків, але і стовпців), що, подібно до GIF, дещо збільшує розмір файлу. На відміну від GIF, де застосовується 100% прозорість, PNG підтримує також напівпрозорі піксели (в діапазоні прозорості від 0 до 99%) за рахунок Альфа-каналу з 256 градаціями сірого.

У файл формату PNG записується інформація про гамму. Гамма є певним числом, що характеризує залежність яскравості світіння екрану монітора від напруги на електродах кінескопа. Це число обчислюється з файлу і дозволяє вводити поправку яскравості при відображенні. Воно потрібне для однакового відображення інформації незалежно від апаратної платформи комп'ютера. PNG підтримується у всіх сучасних браузерях.

PNG-8

PNG-8 — формат подібний до GIF і має покращений формат стиснення даних.

Особливості

- Використовує 8-бітову палітру (256 кольорів) в зображенні, за що і отримав в своїй назві цифру вісім. При цьому можна вибирати, скільки кольорів буде задіяно у файлі — від 2 до 256.
- На відміну від GIF, не відображає анімацію.

Область застосування

Текст, логотипи, ілюстрації з чіткими краями, зображення з градієнтною прозорістю.

PNG-24

PNG-24 — формат, аналогічний до PNG-8, але використовує 24-бітову палітру кольору. Подібно до формату JPEG, зберігає яскравість і відтінки кольорів у фотографіях. Подібно до форматів GIF і PNG-8, зберігає деталі зображення та прозорість.

Особливості

- Використовує понад 16 млн. кольорів, тому, застосовується для повнокольорових зображень.

- Підтримує багаторівневу прозорість, це дозволяє створювати плавний перехід від прозорої області зображення до колірної, так званий градієнт.

- Алгоритм стиснення зберігає всі кольори і піксели в зображенні незмінними. Якщо порівнювати з іншими форматами, то в PNG-24 кінцевий об'єм графічного файлу виходить найбільшим.

Область застосування

Фотографії, малюнки, що містять прозорі ділянки, малюнки з великою кількістю кольорів і чіткими краями зображень.

Векторні веб-формати

SVG. Масштабована векторна графіка

SVG (Scalable Vector Graphics) - це тип векторних файлів, що описують зображення в форматі XML. Формат з'явився в 2001, однак популярність серед веб-розробників він отримав нещодавно, після впровадження підтримки у сучасні браузери. Формат є відкритим стандартом, на відміну від більшості інших форматів, SVG не є чиеюсь власністю.



Файл із зображенням в цьому форматі є звичайним текстовим файлом, який можна відкрити в блокноті і відредагувати. У цьому форматі можна описати не тільки статичну, а й динамічну картинку (анімація), змішати створені вектори з растровою картинкою. Завдяки тому, що кожна фігура для браузера є елементом DOM, за допомогою JavaScript можна описувати досить складні сценарії, взаємодіяти з користувачем.

Розмір об'єктів SVG є меншим за розмір растрових зображень, а самі зображення не втрачають якості при масштабуванні. На відміну від растрових форматів можна взаємодіяти із зображеннями у форматі SVG - за допомогою CSS можна змінювати параметри графіки: колір, прозорість або межі, а за допомогою JavaScript - анімувати зображення.

У SVG є маса функцій, які роблять цей формат рекомендованим для Вебу, особливо якщо SVG використовується для простих зображень типу логотипів, карт, іконок, маркерів.

Переваги формату SVG

- SVG часто важать менше за растрові зображення.

- Формат масштабується, що забезпечує чіткість за будь-якої роздільності екрану.

- SVG-код можна помістити в HTML і налаштувати через CSS.

- SVG-зображення можна анімувати, в тому числі окремі частини, як за допомогою CSS, так і JS.

Втім занадто складні SVG-зображення збільшують розмір файлу. SVG не застосовний до фотографій, тут краще підійдуть формати JPG і webP.

Анімаційні веб-формати

APNG. Анімований PNG

aPNG (Animated Portable Network Graphics) запропоновано в 2004 році компанією Mozilla. Новий стандарт базується на форматі PNG, додано можливість анімації та покращено алгоритми стиснення. Втім компанія, що підтримує формат PNG не визнала новий стандарт, і aPNG не пішов в реліз. Основна причина відмови звучала так «PNG - це формат для нерухомих зображень».



APNG for Chrome™

До 2008 року не було жодних спроб широкого впровадження aPNG. В 2008 році, компанія Mozilla вносить його в свій браузер Firefox, пізніше таку підтримку додано в браузер Opera. З приходом HTML 5, ситуація почала змінюватися. Google, Twitter, Facebook і інші популярні сервіси, стали один за одним відмовлятися від застарілих технологій, і переходити на нові, паралельно, задаючи стандарти розробки.

WebP

<https://developers.google.com/speed/webp/>

Сучасний формат зображень з відкритим вихідним кодом, розроблений компанією Google спеціально для Інтернету, є відгалуженням відеоформату WebM.



Подібно до APNG, він також підтримує 24-бітні зображення, 8-розрядну прозорість. У ньому є стиснення з втратами та без втрат, що у деяких випадках дозволяє досягти невеликого розміру файлів із забезпеченням високої якості, що робить його універсальним. Формат поєднує переваги JPG і PNG без збільшення розміру файлу.

Сьогодні YouTube використовує перетворення мініатюр для відео в формат WebP.

Поки формат підтримується лише браузером Chrome, і має певні недоліки щодо відображення дрібних деталей. Для підтримки в інших браузерах існують способи обходу обмежень (додаткові плагіни або конфігурація), але вони не сприяють використанню формату повсюдно.

FLIF Free Lossless формат зображення

<http://flif.info/>

FLIF - це новий та інноваційний формат зображення, що підтримує анімацію з альфа-прозорістю, прогресивне завантаження (що дозволяє відображати зображення в меншій якості, поки він довантажуються), а також інші можливості у порівнянні з різними форматами. Формат поки не підтримується всіма браузерами, але розробляються JavaScript-плагіни, що дозволяють використовувати FLIF у будь-якому браузері.



Інші формати графічних файлів

PSD (Adobe Photoshop Document)

Внутрішній формат популярного растрового редактора Photoshop останнім часом підтримується великою кількістю програм. Він дозволяє записувати зображення з багатьма прошарками, їх масками, додатковими Альфа-каналами і каналами базових кольорів, контурами та іншою інформацією.



Для стиснення застосовують метод RLE (Run Length Encoding), кодування із змінною довжиною рядка. Дія методу полягає в пошуку однакових пікселів в одному рядку. Якщо в рядку, наприклад, є 3 піксела білого кольору, 21 - чорного, 14 - білого, то застосування RLE надає можливість не запам'ятовувати кожен з них (38 пікселів), а записати як 3 білих, 21 чорний і 14 білих в першому рядку.

Подібно до методу LZW, алгоритм RLE добре працює з штучними і ілюстрованими картинками і гірше з фотографіями. У випадку, якщо фотографія має багато дрібних деталей, RLE може навіть збільшити розмір файлу.

TIFF (Tagged Image File Format)

Апаратно незалежний формат TIFF, на сьогоднішній день є одним з найпоширеніших і надійніших, його підтримують практично всі програми на PC і Macintosh, що пов'язані з графікою. TIFF є кращим вибором при імпорті растрової графіки у векторні програми та видавничі системи. Йому доступно весь діапазон кольорних моделей від монохромної до RGB, CMYK і додаткових кольорів Pantone. TIFF може зберігати контури, Альфа-канали та інші додаткові дані.

The logo for the TIFF format, consisting of the word "tiff." in a bold, lowercase, orange sans-serif font.

TIFF має два різновиди: для Macintosh і PC. Це пов'язано з тим, що процесори Motorola читають і записують числа зліва направо, а процесори Intel - навпаки. Сучасні програми можуть без проблем використовувати обидва варіанти формату. Зазвичай, дані у форматі TIFF не стискаються, але може бути використано LZW-стиснення.

BMP (Windows Device Independent Bitmap)

Рідний формат Windows, який підтримується всіма графічними редакторами, що працюють під управлінням цієї операційної системи. Застосовується, в основному, для збереження растрових зображень, що призначені для використання в Windows. Може зберігати як індексовані (до 256 кольорів), так і RGB-кольори (понад 16 млн. відтінків). Можливе застосування стиснення за алгоритмом RLE.

The logo for the BMP format, consisting of the text ".bmp" in a large, bold, grey sans-serif font.

PDF (Portable Document Format)

PDF запропоновано фірмою Adobe, як платформи-незалежний формат для створення електронної документації, презентацій, передачі верстки та графіки через мережі.



Односторінкові файли PDF відмінної якості може створювати Photoshop. Багатосторінкові PDF можуть створювати програми Adobe Acrobat, PageMaker і програми пакету MS Office.

PDF спочатку проектувався як компактний формат для електронної документації. Тому, всі дані в ньому можуть стискатися, причому, до різного типу інформації застосовуються різні, найбільш прийнятні для них типи стиснення: JPEG, RLE, CCITT, ZIP. Програма Acrobat дозволяє розставляти

гіперпосилання, заповнені поля, додавати у файл PDF відео чи звук, інші дії.

Файл PDF може бути оптимізованим. З нього видаляються елементи, що повторюються, встановлюється посторінковий порядок завантаження сторінок через Веб, з пріоритетом спочатку для тексту, потім для графіки. Якщо елементів, що повторюються, немає, файл, після оптимізації, як правило, дещо збільшується.

PDF найчастіше використовується для передачі по мережах графіки і зверстаного тексту в компактному виді. Він може зберігати всю інформацію для пристрою виведення, яка була в початковому файлі.

WMF (Windows Metafile)

Векторний формат WMF є рідним форматом Windows і використовує його графічну мову. Призначений для передачі векторних даних через буфер обміну (Clipboard). Розпізнається практично всіма програмами Windows, так чи інакше пов'язаними з векторною графікою. Користуватися форматом WMF варто лише у випадках передачі «чистих» векторів. WMF спотворює колір, не зберігає певні параметри, які привласнюються об'єктам в різних векторних редакторах, не містить растрові об'єкти, не розпізнається багатьма програмами на Macintosh.



AI (Adobe Illustrator Document)

Adobe Illustrator – популярний графічний редактор від Adobe. Векторний формат Illustrator можна безпосередньо відкрити у Photoshop, його підтримують майже всі програми Macintosh і Windows так чи інакше пов'язані з векторною графікою і графікою взагалі.



Формат Illustrator є найкращим посередником при передачі векторів з однієї програми в іншу, з PC на Macintosh і навпаки. Втілені або пов'язані з документом растрові файли при обміні через формат Illustrator втрачаються.

CRD (CorelDraw Document)

Векторний формат, що має незаперечне лідерство на платформі PC. Багато програм на PC (FreeHand, Illustrator, PageMaker тощо) можуть імпортувати файли CorelDraw.



У файлах застосовується окрема компресія для векторів і растрів, можуть втілюватися шрифти, файли CorelDraw мають величезне робоче поле 45х45 метрів (цей параметр є важливим для зовнішньої реклами), підтримується багатосторінковість.

Зображення, що створюються програмним шляхом

Є чотири способи створення зображення на веб-сторінці програмним кодом: CSS, SVG, Canvas, JS-анімація.

CSS

Створення зображення на чистому CSS

Замість програмного забезпечення для створення векторних ілюстрацій (такого, як Illustrator, Affinity Designer або Sketch) використовується безпосередньо CSS-код. Створення зображень на CSS може вплинути на креативність мислення і надихнути на підкорення нових вершин цієї мови.



Приклади CSS-зображень:

- <https://a.singlediv.com/>
- <https://habrahabr.ru/post/323086/>

Анімація за допомогою CSS

Властивість CSS - animation дозволяє анімувати переходи від однієї конфігурації стилю CSS до іншої. Анімація складається з двох компонентів: стилю, що описує анімацію і набору ключових кадрів, які вказують початковий і кінцевий стани елемента, а також можливі проміжні точки.

Існують три ключові переваги використання CSS - анімації перед традиційними методами анімації, керованими скриптами:

- Вони прості у використанні для простих анімацій; їх можна створювати не знаючи JavaScript.
- Анімація працює добре, навіть при помірному завантаженні системи. Механізм рендеринга може використовувати пропуски кадрів і інші методи, щоб підтримувати оптимальну продуктивність.
- Надання браузеру управління послідовністю анімації дозволяє йому оптимізувати продуктивність, наприклад, зменшуючи частоту оновлення анімацій, запущених на вкладках, які в даний час не відображаються.

Зображення SVG

SVG - це формат масштабованої векторної графіки. В векторних форматах зберігається не саме зображення, а інструкція з його побудови по точках і кривих.

В растрових форматах інформація про конкретне число точок зображення щільно упакована в бінарний код. В нього марно заглядати і міняти його можна лише в редакторах.

Формат SVG також можна створювати і змінювати в редакторах, наприклад, Illustrator, Sketch або Inkscape. Але ще він текстовий, а значить його можна відкрити в будь-якому текстовому редакторі.



```
<svg width = "20">  
  <rect fill="#fc0" width="20" height="20" />  
  <line stroke="black" x1="0" y1="0" x2="20" y2="20" />  
</svg>
```

При втіленні SVG-файлу, насправді, уставляється не просто картинка, а ціла сторінка. Зі своєю системою координат, стилями, скриптами і особливостями. Існує чотири основних способів втілення SVG-зображень зі своїми особливостями.

Уставляння в елементі `img` HTML-коду

Це самий ефективний спосіб завантажити будь-яку картинку.

```

```

В такому SVG-зображенні не працюватимуть скрипти і будь-які спроби взаємодії з елементами всередині не є успішними. Файл буде як за склом: дивитися можна, а чіпати не можна. Хоча всередині все інше прекрасно працює, включаючи CSS-анімації.

Такий спосіб найкраще підходить для контентних зображень, яким не потрібна взаємодія: логотипи, графіки, схеми.

Фонові картинка в CSS

Уставляння основною картинкою до елемента, псевдоелементу або контенту. В такому SVG-зображенні також не можна змінювати оригінальний вміст.

```
.picture {background-image:url (picture.svg);}
```

Цей спосіб підходить для оздоблювальної графіки, для якої не потрібна взаємодія: фони, іконки та інші нескладні зображення.

Уставляння через тег <object>

Третій спосіб, уставляння через тег object. У цьому випадку працюють скрипти, взаємодія, анімація - якщо вони описані всередині SVG. Між тегами object можна вставити код, що повідомить користувача про використання застарілої версії браузера.

```
<object type = "image/svg + xml" data="picture.svg">
  <img src = "picture.png">
</object>
```

Замість тегу object можна використовувати тег iframe, що буде сприймати SVG-код як іншу сторінку. Але object працює краще і підлаштовується під розміри картинки.

За гнучкість доводиться платити: через те, що це вже не просто графіка і там можна додавати скрипти, до такого способу висуваються інші вимоги безпеки. Наприклад, картинку з іншого домену просто так вже не вставити.

Цей спосіб підходить, коли потрібно додати інтерактивну графіку: ігри, графіки, діаграми.

Пряме уставляння вмісту SVG-файлів на сторінку

Цей спосіб став можливим при впровадженні стандартів HTML5 і вміст SVG-файлів тепер можна вставляти прямо на сторінку, як будь-які інші теги.

```
<h1>Квадрат</h1>
<svg>
  <rect fill="#fc0" width="120" height="120" />
</svg>
```

З таким SVG-елементом можна маніпулювати як зі звичайними HTML-елементами: стилі, скрипти. Наприклад, можна змінювати колір заповнення при наведенні і описувати все в загальних стилях.

```
<style>
rect:hover {fill:#090;}
</style>
```

Нажаль, такі зображення не можна зберігати.

Формат SVG є значно більшим, ніж графічний формат, і є достатньо матеріалів, які розкривають його можливості

- Статті Сарі Суайдан — <https://sarasoueidan.com/tags/svg/>

- Курс «Оздоблення SVG-фігур» — <https://htmlacademy.ru/courses/187>

- Курс «Знайомство з SVG» — <https://htmlacademy.ru/courses/130>

- SVG vs PNG — <https://svgvspng.com/#ru>

Canvas

HTML елемент `<canvas>` створює область на веб-сторінці, яку можна використати для відтворення графіки в режимі реального часу за допомогою сценаріїв (наприклад, за допомогою мови JavaScript). Він може бути використаний для створення онлайн-ігор, годинників, анімації тощо.



Додавання на сторінку

Елемент `<canvas>` є звичайним HTML-елементом:

```
<canvas id="test" width="600" height="200"> </canvas>
```

Цей елемент має три атрибути: `id`, `width` і `height`. Атрибут `id` (ідентифікатор) надає елементу унікальне ім'я - `test`, це потрібно для того, щоб мова сценаріїв могла отримати до нього доступ. Атрибут `width` визначає кількість пікселів по горизонталі, яке `<canvas>` буде займати на веб-сторінці. Аналогічним чином `height` визначає вертикальну займану область. Таким чином, браузер зможе виділити простір потрібного розміру на сторінці, відповідно до зазначених значень.

За замовчуванням полотно є прозорим і не має рамки, тобто є порожнім простором у вікні браузера, в якому можна малювати.

Використання CSS для вказування розміру

За замовчуванням елемент `canvas` має ширину 300 пікселів і висоту 150 пікселів. Якщо не поставити атрибути `width` і `height`, то елемент отримає розміри за замовчуванням. Якщо потім задати розміри, використовуючи CSS, припустимо, 600x200 пікселів, то `<canvas>` розміром 300x150 пікселів піддається масштабуванню відповідно до заданих параметрів, так само і все, що буде в ньому намальовано.

При використанні атрибутів `width` і `height` можна задавати розміри для елемента, які будуть більше (або менше), ніж 300x150 пікселів, і все створене в ньому буде відображатися нормально. Таким чином, використовувати CSS для налаштування розміру буде доцільно, якщо потрібно дійсно масштабувати елемент.

Робота зі старими версіями браузерів

Не всі браузери підтримують елемент `<canvas>`, тому, для інформування користувачів, чиї браузери не підтримують його, можна помістити спеціальне повідомлення або альтернативний вміст, який буде відображатися тільки в тому випадку, якщо елемент не підтримується, в іншому випадку проігнорується:

```
<! - Розміщуємо інформацію для користувачів ->
```

```
<canvas>
```

```
    Ваш браузер не підтримує елемент <canvas>
```

```
</canvas>
```

```
<! - або можна помістити картинку рівну розміру елемента ->
```

```
<canvas>
```

```
    
```

```
</canvas>
```

Атрибути

- **height:** Визначає висоту елемента в пікселях (наприклад: "400"). Значення за замовчуванням - "150".

- **width:** Визначає ширину елемента в пікселях (наприклад: "150"). Значення за замовчуванням - "300".

```
<canvas id="myCanvas" width="400" height="150"></canvas>
```

```
<script type="text/javascript">
```

```
    var canvas=document.getElementById('myCanvas');
```

```
    var ctx=canvas.getContext('2d');
```

```
        ctx.fillStyle='#CC5200';
```

```
        ctx.fillRect(0,0,100,100);
```

```
</script>
```

Елемент `<canvas>` підтримує [Глобальні атрибути](#) и [Події](#)

2. Мультимедіа у Веб

Дизайн - це більше ніж візуальне представлення даних, це управління взаємодією користувача і веб-сторінки. Як результат, мультимедійні об'єкти відіграють надзвичайно важливу роль в справі передачі інформації.

Проектуючи сторінки, необхідно з самого початку пам'ятати про інтерактивну природу веб-простору і сприймати її прояви як природну частину дизайну.

HTML5 є платформою для реалізації не просто сторінок, а веб-додатків, що містять анімацію, графіку, відео і аудіо. Все це робить HTML однією з найпопулярніших мов, яка динамічно розвивається і вдосконалюється.

Веб-анімація

Взаємодія користувачів з сучасними веб-сайтами серйозно зав'язано на анімації. Якісна і доречна анімація пожвавлює сторінку і сприяє кращому сприйняттю інформації. Вона здатна повідомляти про певні стани сторінки та привертати увагу. Анімація допомагає користувачеві побачити результат його дій і може впливати на його поведінку.

Анімація і послідовності подій

Анімація при завантаженні даних



Такий прийом створює відчуття того, що дія виконується швидше, ніж насправді. Анімована картинка свідчить, що прогрес завантаження відбувається і відвідувач легше сприймає цей час. Чим простіше анімація завантаження - тим краще. При цьому слід дуже обережно ставитися до будь-яких додаткових ефектів, на зразок звуків. Зазвичай вони просто не потрібні.

- <https://tobiasahlin.com/spinkit/>
- <https://icons8.com/cssload/en/horizontal-bars/>

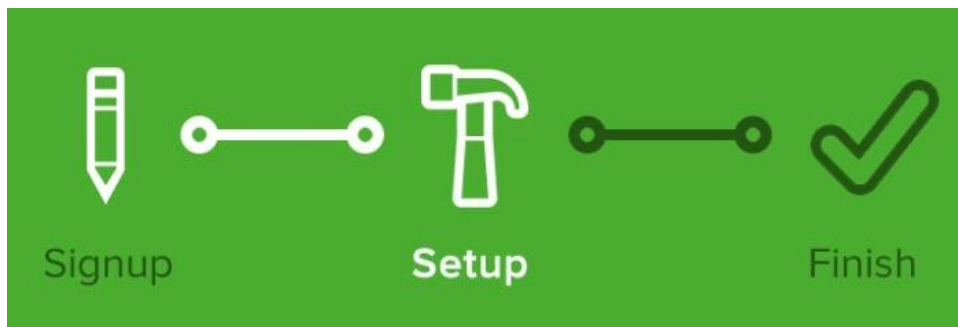
Анімація процесів

Анімація може показати виконання певної лінійної послідовності дій, що не залежить від користувача. Типовим прикладом є прогрес завантаження. (<https://tickets.ua/>)



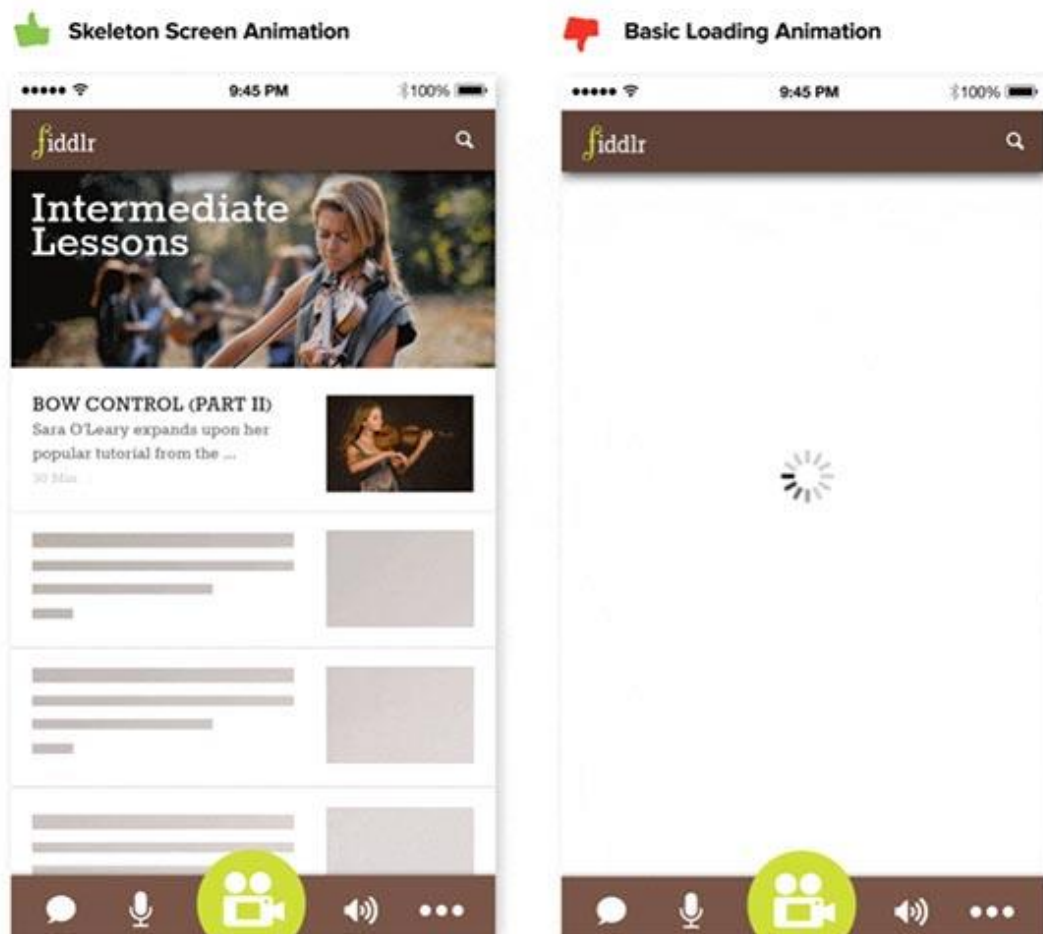
Анімація покрокових операцій

Анімацію можна використовувати для процесів, які передбачають виконання користувачем покрокових операцій, щоб відображати послідовність виконаних дій.



Анімація і каркасне відображення веб-сторінок

Варіант каркасного відображення веб-сторінок передбачає наявність простору, яке поступово заповнюється завантаженою інформацією. Поступове, не розірване періодами «порожнечі», наповнення сторінки, дає відчуття швидкості виконання дій. Цю техніку можна використовувати на практично будь-якому сайті разом з ненав'язливою анімацією процесу завантаження, що забезпечує утримання уваги користувача.



Каркасні екрани завантажують елементи користувацького інтерфейсу ще до того, як буде повністю відображено вміст сторінки.

Візуальний зворотний зв'язок

Анімована реакція на дії користувача (onboarding)

Хороший дизайн взаємодій з користувачем передбачає зворотний зв'язок, що повідомляє про результати взаємодії, робить його видимим і зрозумілим. Якщо відвідувач сайту не знає, з якими саме елементами сторінок можна працювати, чого від них можна очікувати, він буде відчувати себе розгубленим. Потрібно продумано спроектувати варіанти взаємодії, зробивши їх чіткими і зрозумілими.

Популярним стає вживання онбордінгу, це загальна назва підходів, що допоможе користувачеві ознайомитися з сайтом, сервісом, продуктом. До поширених типів онбордінгу відносяться спливаючі вікна (popup) і підказки (tooltips), які супроводжують користувачів, поки ті освоюють можливості сайту. Онбордінг допомагає акцентувати увагу на важливих кнопках, вкладках і інших елементах навігації, демонструє функціонал і цінність товару, послуги або пропозиції. Розрізняють два основних етапи онбордінгу:

- Перший етап називається first-use, що застосовується, якщо користувач вперше зайшов на ресурс. На цьому етапі знайомства користувач повинен зрозуміти: чи може продукт вирішити його проблему чи підходить він конкретно йому.
- Другий етап включається, коли користувач вже користується продуктом. Його мета - навчити користувача новим функціям, утримати і мотивувати підписуватися далі, бути активним, рекомендувати, ділитися результатами своєї роботи в додатку.

<https://www.duolingo.com/>

Анімація елементів управління для настільних і мобільних сайтів



Один з найпоширеніших прикладів візуального зворотного зв'язку - анімація при наведенні курсору на інтерактивний елемент.

Якщо користувач не впевнений в призначенні елемента управління, він намагається навести на нього курсор миші. Анімація елемента при наведенні курсору, на

інтуїтивному рівні, повідомить користувача про те, що з об'єктом можна взаємодіяти.

На мобільних сайтах, кнопки та інші інтерактивні елементи повинні бути оснащені візуальними підказками. Підказки повинні вказувати на те, як взаємодіяти до того, як користувач торкнеться елемента. Після торкання потрібно надати візуальний зворотний зв'язок, щоб засвідчити реакцію системи.

<https://www.whitelabeldevelopers.ru/articles/20-udivitelnyix-animirovannyix-knopok-na-chistom-css>

Привертання уваги за допомогою анімації

Зміна стану елементів на сторінці скеровує увагу користувача і підкріплює виконуваним дії. Наприклад, при введенні даних в форму, поле підсвічується і користувач розуміє, що потрібно вводити текст.

Переходи і зміна станів

Плавна зміна станів

Анімація переходів надає можливість плавно змінити колір фону, прозорість блоку, затемнення. Без анімації подібні зміни будуть з різким переходом, який відчувається як неприродний «стрибок». За допомогою анімації можна забезпечити плавні переходи між екранами. Добре спроектовані переходи дають користувачам чітке розуміння того, на чому слід зосередитися.

Довгий скрол

Раніше дизайнери прагнули заповнити верхню область сторінки цінною інформацією. Тепер же «правило верхньої половини» втратило абсолютну істинність. Дизайнери застосовують красиві плавні ефекти для скролінгу сторінки як вниз, так і вгору. Це робить прокрутку простою і зручною.

Анімації оживляють процес скролінгу і приносять задоволення користувачеві.

Типи анімацій

Анімація є важливою складовою дизайну веб-сайту і у веб-дизайнера є широкі можливості вибору багатьох технологій, які допоможуть в досягненні конкретних цілей, що ставляться перед сайтом. Тип анімації і програмного забезпечення, необхідного для її створення визначається в залежності від конкретного призначення.

GIF-анімація

Анімація з використанням формату GIF (Graphic Interchange Format) є простою в реалізації і відтворенні на сторінках. Анімовані GIF-зображення працюють як традиційна покадрова анімація, яку застосовують у мультфільмах. Анімоване зображення містить певне число кадрів, де кожен кадр є частиною анімації, також задаються параметри, які дозволяють регулювати швидкість і тривалість анімації. Програма, за допомогою якої створюється GIF-анімація (наприклад, PhotoShop) зберігає кадри та відмічені опції в GIF-файлі, а браузер інтерпретує задані параметри і відтворює анімацію, створюючи зоровий образ руху.

<https://west-info.ua/>

Веб-засоби для створення анімацій: CSS і JavaScript

Однією з важливих особливостей HTML5, на сьогоднішній день, є можливість створення анімації. Хоча HTML є достатньо простою мовою, процес створення анімації вимагає не мало зусиль, навичок, а іноді і поглибленого вивчення JavaScript і CSS. Втім, щоб полегшити цей процес можна скористатися спеціальними інструментами для створення HTML5 анімації. Кожен з них має власні переваги, тому можна вибрати те, що найбільш задовольнятиме вимогам.

У веб-розробці на сьогодні популярними є два основних підходи для створення анімації: за допомогою CSS і JavaScript. Вибір залежить від характеристик проекту, а також від того, яких ефектів прагне досягти розробник.

Анімація за допомогою CSS

<https://webdesign.tutsplus.com/ru/articles/pure-css-animation-inspiration-on-codepen--cms-30875>

<https://icons8.com/cssload/en/horizontal-bars>

Анімація за допомогою CSS, без сумніву, є найпростішим способом змусити щось рухатися на екрані. Окрім використання переходів, можна контролювати окремі ключові кадри, тривалість і проходи анімації. Анімація за допомогою CSS досі залежить від браузера, тому, для її позначення потрібно вказувати префікси.

Переходи і анімація, що виконані засобами CSS, ідеально підходять для випадного меню або спливаючої підказки. JavaScript можна використовувати для контролю стану, проте, сама анімація буде виконана засобами CSS.

Анімація за допомогою JavaScript

<http://beloweb.ru/novichkam/10-krutyih-primerov-raboty-html5-canvas.html>

Створювати анімацію за допомогою JavaScript складніше, ніж писати переходи або анімацію засобами CSS, проте, JavaScript, як правило, надає розробнику ширші можливості. Зазвичай, використовується функція `requestAnimationFrame`, а потім на кожному кадрі анімації вручну визначається значення кожної властивості анімованого елемента.

Оскільки анімація засобами JavaScript дозволяє повністю контролювати стилі елементів на кожному кроці, анімацію можна уповільнювати, зупиняти, відтворювати в зворотному напрямку і виконувати інші маніпуляції.

JavaScript бібліотеки для анімації

jQuery Animate

jQuery Animate - jQuery-анімація з використанням `Animate.css`. Бібліотека для роботи з CSS-анімацією. Раніше не містила шаблонів jQuery і обмежувалася лише CSS. На сьогодні доповнена скриптом `jAnimate`, що підсилює бібліотеку `Animate.css` можливостями JQuery і дозволяє використовувати анімацію для різних елементів сайту.

Chart.js

Бібліотека дозволяє створювати різні діаграми та графіки: лінійні діаграми, гістограми, діаграми з областями та інші. `Chart.js` проста у використанні, на офіційному сайті надано докладну документацію.

Незважаючи на те, що бібліотека пропонує лише 6 видів діаграм, вона добре підійде для невеликих проектів. Програма використовує HTML5 Canvas для побудови діаграм, створює простий дизайн, що швидко реагує на зміни, і стрімко перетворюється в одну з найпопулярніших відкритих бібліотек з діаграмами.

Bubble.js

Bubble - JQuery плагін для створення анімованих підказок. Підказки використовуються для звернення до відвідувачів, яким може знадобитися допомога. Використовуючи даний плагін, можна легко виділити область сайту і додати для неї підказку. Можна створювати «майстрів» процесів, які будуть виводити численні

спливаючі підказки, щоб пояснити конкретну функцію. Прості налаштування тривалості анімації за допомогою JS і CSS допоможуть створити власні користувацькі теми.

Talos.js

Talos - jQuery-плагін для створення анімованих горизонтальних підміну з використанням JQuery, CSS3 і HTML5. Ефекти анімації створюються за допомогою властивостей анімації і переходів CSS3, вони досить просто можуть бути відредаговані. Чотири унікальних ефекти анімації, сім тем, оптимізація під мобільні пристрої (адаптивний дизайн), ефекти наведення, безкоштовні шрифти та багато іншого.

Notifications Pro

Push-повідомлення стають популярними, оскільки дозволяють легко привернути увагу користувача і донести до нього певну інформацію. Особливо це стосується бізнес-сайтів, для яких вони можуть стати орошим способом дати користувачам знати, що відбувається. Плагін підтримує вісім різних позицій, щоб відображати оповіщення - в основному в різних кутах сайту.

Difference.js

Difference - jQuery-плагін для створення зворотних відліків. Модулі та бібліотеки для створення зворотних відліків найкраще підходять для тих випадків, коли потрібно анонсувати запуск певної сторінки або реліз конкретного продукту. Difference найкраще підходить для цього.

Плагін підтримує дві теми і шість колірних схем для п'яти різних форм зворотного відліку і сотні можливих комбінацій. Простий і потужний Difference дозволяє встановити інтервали зворотного відліку (дату і час початку та закінчення), тип зворотного відліку, теми, кольори для кожного окремого поля і розміри.

Lush.js - контент-слайдер

JQuery-плагін контент-слайдера для створення унікальної анімації з використанням властивостей переходів і анімації CSS3. Використання оригінальної мови дозволяє підвищити продуктивність, створювати стильний зовнішній вигляд віджетів.

Сінемаграфія - живі фотографії

<https://videoinfographica.com/cinemagraph/>

Сінемаграфія - це нова концепція фотографії, яка об'єднує відео і фото. Сінемаграфічний ефект утворюється з відео, в якому певні частини кадру залишають статичними, а в інших простежується рух. Природно, що такі роботи можуть бути затребувані лише в веб-виданнях і самої вдячної нішею їх використання звичайно ж є реклама. Сінемаграфія відтісняє flash банери, оскільки вихідним форматом такого живого фото є GIF, який не потребує додаткових плагинів і ресурсів.

Ефект сінемаграфії може бути створений як в Adobe Photoshop так і в інших спеціальних програмах. Зараз розроблено досить багато програм під iPhone і Android, що дозволяють робити з відео фото з сінемаграфією. Принцип їх роботи однаковий - з обраного невеликого шматочка відео з бажано зацикленним рухом певного об'єкту виокремлюється та частина, яка буде рухатися, а решта залишаються незмінними як в першому кадрі.

Сінемаграфія - це досить складне мистецтво. По-перше, потрібна концепція і та особливість, яка буде привертати увагу глядача. По-друге, відео для сінемаграфії має бути знято за певними правилами. І по-третє, сама пост-обробка, яка вимагає кропіткої роботи та уваги. Як правило, для ефекту сінемаграфії, автори використовують такий рух як: погойдування на вітрі, волосся що розвиваються, потік води, рух тіні при нерухомості самого об'єкта і багато іншого.

Звук в Інтернет

Веб довгий час був безмовним середовищем, але на сьогодні звук є дуже популярним. Продуманий звуковий супровід допоможе відвідувачам сайту отримати максимум вражень від сторінки і утримати їх на сайті.

Перші звукові формати дозволяли створювати файли з настільки великими розмірами, що на завантаження хвилиного музичного фрагмента потрібно було очікувати хвилинами, а то і годинами.

Технології розвиваються і веб-розробники розуміють, наскільки доречним є звук. Використання звуку внесло зміни до дизайну і конструкції сайтів. Наприклад, тепер на деяких сайтах ведеться радіомовлення (webcast — співзвучно до broadcast - віщати, передавати теле- або радіосигнали). Тисячі сайтів в Веб віщають в прямому ефірі. Webcast широко використовують - від

музики до прямих трансляцій президентських мов і спортивних ігор.

Якщо сайт має звуковий супровід, слід передбачити кнопку, яка дозволить користувачеві швидко вимкнути музику або звук.

Подібно до графічних файлів, аудіофайли теж мають власні формати. Загальноприйнятими форматами звукових файлів, які добре працюють на платформі Windows є формати WAV, AU, SND, MP3 і MIDI. На платформі Macintosh застосовують формати AIF, AIFF, SND, AU, MP3 і MIDI. Деякі з форматів (AU, SND, MP3 і MIDI) є універсальними і можуть використовуватися як в Windows, так і в Macintosh.

Відео на Веб-сторінці

Популярною на сьогодні областю веб-дизайну є втілення відео на веб-сторінках. Застосування відео в Інтернет постійно зростає, тому стрімко вдосконалюються технології, що дозволяють швидко завантажувати відеофайли. Уміло застосовані відео можуть зробити веб-сайт більш привабливим для користувачів.

Популярним трендом в веб-дизайні є великі відео на задньому плані веб-сайтів. Відео-фони можуть відтворюватися плавно і не гальмувати на відміну від фонів, що створені за допомогою JavaScript, які вимагають завантажити певний код і зображення перш ніж користувач побачить результат анімації.

Найбільш популярним способом створення відео-фону є використання HTML5 тегу <video> разом з деякими параметрами CSS. Варто звернути увагу на альтернативні джерела відео, наприклад, YouTube. В цьому випадку, можливим є повільні відповіді сервера, оскільки запит буде відправлятися в YouTube.

Віртуальні екскурсії

Віртуальної екскурсії є доволі ефективним способом подачі інформації і надання відвідувачам можливості ознайомитися з вмістом.

Однією з технологій створення віртуальної реальності в Інтернеті є панорамні зображення. Компонентами цієї технології виступають панорами і об'єкти, які об'єднуються в сцени. Панорамні фільми надають можливість кругового огляду, дозволяють збільшити або зменшити зображення. При зміні точки зору відбувається автоматичне коректування перспективи.

Об'єднання панорам і об'єктів в багатовузловій сцени дозволяє створювати віртуальний світ, який можна досліджувати переміщенням від одного вузла до іншого. Такі сцени зберігаються як у вигляді одного файлу, так і в кількох взаємозв'язаних файлах, де віртуальна сцена може бути об'єднана із звуковою мультимедійною інформацією.

WebAR

Доповнена реальність стає популярною серед розробників та користувачів смартфонів. Щоб випробувати AR, користувачі повинні встановити спеціальний додаток, але часто видаляють його після кількох використань або навіть не завантажують його взагалі. WebAR покликаний стати вирішенням проблеми небажання користувачів випробувати його через програми, і є досить вагомими докази того, що це дійсно вдасться.

WebAR – це реалізація доповненої реальності, доступ до якої здійснюється через веб-браузер, а не через програму. Він забезпечує цей веб-інтерфейс AR з використанням таких технологій, як WebGL, WebRTC, WebVR та API. Для доступу до нього просто потрібний телефон.

WebAR дозволяє користувачам смартфонів відкривати для себе технологію AR найпростішим способом – через Інтернет – без необхідності встановлення. Він просто долає бар'єри та створює інтерактивні 3D-моделі, до яких можна отримати доступ через QR-код або посилання. На даний момент WebAR пропонує обмежену функціональність порівняно з AR на основі додатків, але він вже містить просту анімацію, відео і певний ступінь інтерактивності. WebAR підтримує виявлення цілей зображення.

WebAR є частиною іммерсивного Інтернету і, хоча для нього не потрібна програма, він все ж таки має деякі технічні вимоги. Насамперед, смартфон повинен мати датчики, такі як гіроскоп, акселерометр та RGB-камеру – речі, якими оснащено більшість сучасних смартфонів – для запуску сторінок WebAR. Браузер повинен підтримувати WebXR, API, який дозволяє користувачам переглядати контент AR/VR без встановлення додаткових плагінів або програмного забезпечення.

WebAR як AR, в цілому, справді може дати дивовижний досвід для своїх користувачів, але є багато аспектів, які слід продумати під час його розробки. Щоб WebAR був

функціональним та зручним для користувача, він повинен працювати таким чином.

- По-перше, слід визначити, де знаходиться пристрій у тривимірному просторі – його становище та орієнтація. Цей крок необхідний для синхронізації 3D-зображення поверх реального світу. Цей процес іноді називають шістьма ступенями свободи, що означає здатність відстежувати три осі положення та три осі орієнтації.

- По-друге, необхідно експонувати потік камери, її поле зору та перспективу камери. Ця функція також потрібна для синхронізації віртуального та реального світу.

- По-третє, для безпомилкового запуску AR у веб-версії необхідно включити функцію розуміння сцени, тобто здатність пристрою знаходити поверхню, на яку можна помістити 3D-об'єкт, та оцінювати світло у навколишньому середовищі.

Звичайна доповнена реальність на основі програм вже широко використовується та доступна для всіх. Такі програми, як Snapchat або Instagram, мають маски, що стали справжнім трендом. Ікеа та LEGO також успішно впровадили цю технологію. WebAR має всі шанси стати таким же популярним, як і додаток на основі додатків, або навіть замінити його.

Можливі реалізації WebAR у бізнесі, маркетингу чи повсякденному житті.

- Перший і, можливо, найбільш захоплюючий додаток WebAR може бути для пошукових запитів. Google вже оголосив, що додасть доповнену реальність до пошукової системи. Наприклад, якщо шукати інформацію про акули, можна помістити анімовану версію тварини поверх реального фону; або можна приміряти туфлі, щоб побачити, як вони будуть виглядати насправді. Остання функція відкриває абсолютно новий діапазон можливостей для покупок в Інтернеті, оскільки клієнти будуть більш впевнені у своїй покупці, якщо вони зможуть спершу випробувати її, взаємодіючи з 3D-моделлю продукту.

- Електронна комерція та реклама. WebAR може підняти досвід електронної комерції нового рівня. У разі маркетингу він пропонує нову платформу для рекламних кампаній, оскільки люди можуть просто відсканувати QR-код або перейти за посиланням і

отримати цікавішу та інтерактивну рекламу, яка їм дійсно сподобається.

- Освіта. Застосування WebAR в освітньому процесі може стати відмінним способом залучити учнів до навчальних класів та зробити весь навчальний процес більш інтерактивним та захоплюючим. Крім того, це може допомогти студентам орієнтуватися у кампусі.

- Розважальна програма. Використання WebAR у розважальних цілях, можливо, найбільш очевидне, оскільки взаємодія з ним сама по собі справді весела. Його можна використовувати для ігор або навіть просування фільмів. Sony Pictures фактично використала WebAR для просування Джуманджі, і вони запропонували справді захоплюючий досвід, заснований на голосовій взаємодії, щоб подарувати захоплюючу 3D пригоду.

- Бізнес. Одна з найцікавіших реалізацій WebAR – візитки у 3D AR. Під час сканування QR-коду на екрані телефону відображається інтерактивна картка. Звідти можна переміщатися соціальними мережами, електронною поштою людини. Візитна картка WebAR також дозволить відслідковувати аналітику: скільки людей використовували QR-код.

WebAR скоро стане мейнстримом і, ймовірно, найбільше використовуватиметься маркетологами. Найкраще в WebAR це те, що він може створити надзвичайно простий спосіб доступу до інформації. Можливість інтегрувати інформацію в будь-якому місці середовища може створити простір, де дані будуть доступні без пошуку, за одне сканування QR-коду або один дотик. Знайти свій шлях стане простіше завдяки інтерактивним картам, що накладають візуальні вказівки на реальне тло.

Найімовірніша розробка найближчим часом – це найкраща адаптація AR до веб-формату, що означає найкраще розуміння сцени та створення хмарного простору для спільного використання WebAR. Подальший розвиток WebAR також може бути прискорено за рахунок широкого визнання в суспільстві гарнітур, які можуть дати зовсім інший досвід AR. Це означає миттєвий доступ до інформації, коли з'являється контекстно-релевантний контент.

Перенесення доповненої реальності до Інтернету може бути найкращим варіантом для забезпечення майбутнього цієї технології. Звільняючи користувачів смартфонів від необхідності

встановлювати численні програми, WebAR може зробити доповнену реальність набагато доступнішою для них та вигіднішою для бізнесу. Оцифрування продуктів за допомогою WebAR відкриває нову еру у відносинах із покупцем та значно збагачує бізнес-стратегію брендів.

ТЕМА 5. СЛУЖБА DNS

1. Домени верхнього рівня

- **Міжнародні домени загального користування (gTLD - generic Top-Level Domain).** Загальні домени верхнього рівня, що керуються організацією ICANN.

- **Національні домени (ccTLD - CountryCode Top-Level Domain).** Національні домени верхнього рівня (ccTLD) делеговані до відповідних національних реєстраторів, які встановлюють правила реєстрації в них або самі, або згідно з вказівками уряду. Знаходяться під контролем IANA.

- **Інтернаціоналізовані домени (IDN - Internationalized Domain Names).** Доменні імена, які містять символи національних алфавітів. Умовно ці домени можна поділити на часткові **IDN SLD (Second-Level Domain)** і повні **IDN TLD (Top-Level Domain)**. IDN верхнього рівня керуються і знаходяться під контролем ICANN.

- **Національні домени верхнього рівня (NLIA Native Language Internet Address)** – це є повні інтернаціоналізовані домени IDN TLD (Top-Level Domain).

- **Нові домени верхнього рівня (New gTLDs).** Знаходяться під контролем IANA.

- **Службові зарезервовані домени (iTLD).** Доменні імена, що зарезервовані для різних цілей і не використовуються як реальні імена доменів в глобальній службі DNS. Зарезервовані імена є недоступними для реєстрації.

Міжнародні домени загального користування (gTLD)

gTLD - Generic Top Level Domains, родові домени. Спадщина епохи раннього Інтернету.

Від початку розвитку Інтернету, коли мережа поширювалася лише на теренах США, було створено перші 6 доменів:

1984 р

- .net (Network) Організації, що забезпечують роботу мереж.
- .mil (Military) Військові організації.
- .gov (Government) Урядові організації.
- .edu (Educational) Освітні заклади.

- .com (Commercial) Комерційні структури.
 - .org (Organization) Некомерційні організації.
- 1988 р
- .int - для міжнародних організацій.
- 2000 р.
- .biz (Business Organizations) - тільки комерційні організації (з обмеженнями).
 - .info (Information) - домен відкритий для всіх.
 - .name (Personal) - для персональних сайтів (з обмеженнями).
 - .pro (Professionals) - для фахівців певних професій (з обмеженнями).
 - .aero - організації та фізичні особи, так чи інакше пов'язані з аеро індустрією.
 - .museum - для музеїв
 - .coop - для кооперативних організацій
- 2001 р.
- .arpa - службовий домен, технічна підтримка глобальної мережі.
- З 2005 р.
- .asia - домен для резидентів азіатсько-тихоокеанського регіону.
 - .cat - призначений для представників каталонського лінгвістичного та культурного співтовариства.
 - .eco - для ресурсів, пов'язаних з екологією.
 - .jobs - для сайтів з інформацією про затребувані професії і вакансії.
 - .mobi - для сайтів і сервісів, орієнтованих на роботу з мобільними телефонами і безпроводними пристроями.
 - .post - для поштових організацій.
 - .rich - для багатих.
 - .tel - для зберігання і управління персональними і корпоративними контактами.
 - .travel - для туристичної індустрії.
 - .xxx – для сайтів з порнографічним вмістом.

Список адрес для кожного родового домену верхнього рівня підтримує одна реєстратура, наприклад, домен .com адмініструється компанією VeriSign. Функцію «продавців» беруть на себе реєстратори. ICANN здійснює загальну координацію

системи DNS, укладаючи угоди і видаючи акредитацію реєстратурам та реєстраторам. Ця організація призначає оптову ціну, за якою реєстратура (наприклад, VeriSign) здає в оренду реєстраторам доменні імена, і встановлює певні умови надання послуг, що надаються реєстратурою і реєстраторами.

Основна група доменів верхнього рівня загального користування складається з доменів COM, INFO, NET, і ORG. Домени BIZ, NAME і PRO також вважаються доменами верхнього рівня загального користування. Однак ці домени мають деякі обмеження на реєстрацію.

До доменів верхнього рівня загального користування відносять домени, які створено на ранньому етапі розвитку системи доменних імен та призначаються для конкретних установ чи організацій. До таких доменів відносяться EDU, GOV, INT і MIL.

Національні домени верхнього рівня ccTLD

Національні домени верхнього (першого) рівня - це домени, що виділені корпорацією ICANN для використання в конкретних країнах.

Існує кілька умов для реєстрації національних доменів другого рівня. Найголовнішим умовами є визнання країни світовою спільнотою. Доменне ім'я національного домену має відповідати двохсимвольним кодам країни згідно зі списком міжнародного стандарту ISO 3166. Трьохсимвольні коди можуть використовуватися у разі необхідності запобігання можливих непорозумінь і конфліктів.

Національні домени 2 рівня призначені для використання всередині відповідної держави. Більшість Адміністраторів національних доменів дозволяють реєструвати домени лише резидентам відповідної країни (громадянам або юридичним особам, що зареєстровані або мають філію на території даної держави). Однак існують і винятки. На даний момент делеговано 253 національних доменів 1 рівня (ccTLD).

Від початку національними доменами керували різні інституції, які отримали акредитацію на початкових етапах розвитку Інтернету, коли уряди не цікавилися такими питаннями. В їх числі наукові установи, технічні асоціації, неурядові організації і навіть приватні особи. В багатьох випадках право на управління

кодами країн видавалося за тим же принципом «першим прийшов - першим обслужили».

Тепер, повноваження з управління національними доменами передаються Інтернет-спільнотам країн. В якості адміністраторів національних доменів найчастіше виступають некомерційні організації або урядові структури. Як правило, адміністратори національних доменів займаються лише загальним керівництвом, а право безпосередньої реєстрації доменних імен в національній зоні надається компаніям-реєстраторам доменних імен.

Залежно від правил, що встановлюються адміністраторами національних ccTLD, домени країн підрозділяють на закриті та відкриті.

- В закритих національних доменах реєстрантами можуть бути лише громадяни цієї країни чи організації-резиденти. Такі домени, зазвичай, розвиваються досить повільно (наприклад, домени Швеції .SE та Ірландії .IE).

- У відкритих національних доменах правила є більш демократичними і їх розвиток відбувається швидше. До такого виду національних доменних зон відносяться, наприклад, домен Німеччини .DE, домен Китаю .CN, Росії - .RU.

- Існують також ccTLD з частковими обмеженнями на реєстрацію, які можна віднести до умовно відкритих (в домені .EE (Естонія) на одну людину можна зареєструвати лише одну адресу).

За кількістю зареєстрованих доменів серед всіх національних доменних зон перше місце належить Німеччині: у зоні .DE близько 12 млн. імен. Друге місце займає домен Китаю .CN: число зареєстрованих адрес більше за 10 млн. Третьою в цьому списку є доменна зона Великобританії .UK з 6,5 млн. доменних імен. Число доменів в Українській зоні .UA складає приблизно 600 тис. (<https://hostmaster.ua/UAstat/>)

Регіонально-тематичні доменні зони

У більшості ccTLD існують домени другого рівня, що дублюють деякі тематичні зони, наприклад: .net.ua, .com.ua, .biz.ua.

Регіонально-тематичні зони слід використовувати, коли ресурс точно відповідає одній з тематичних зон, однак не призначений для міжнародного масштабу, наприклад .gov - уряд США, .gov.ua - уряд України.

Регіонально-тематичні зони можуть мати свої обмеження на реєстрацію: наприклад, .edu.ua дозволяє реєструватися навчальним закладам України, але лише вищим, причому не нижче третього рівня акредитації.

В деяких країнах регіонально-тематичні доменні зони можуть мати дещо змінені назви: наприклад, у Великобританії замість .com.uk використовується .co.uk.

Субрегіональні доменні зони

Ці домени другого рівня призначаються для окремих регіонів країни. Наприклад, .lviv.ua - Львівська область України; .kuiv.ua – місто Київ. Домен варто реєструвати в субрегіональній зоні, якщо орієнтованість ресурсу на регіон значно переважає тематичну.

Принцип вибору приблизно такий же, як і для територіальної зони, тільки зі звуженням до окремого регіону країни.

Інтернаціоналізовані домени IDN

Інтернет зародився в англomовному середовищі і обмеженням існуючої системи доменних імен стала необхідність використовувати лише 37 символів ASCII - латинські букви від А до Z, цифри і символ «-». Швидке зростання кількості користувачів Інтернет в світі призвело до появи доменних імен (від 2 рівня), які зазначено не латиницею, а символами національного алфавіту.

- Це зручніше, бо в доменному імені можна використовувати національну мову.
- Вирішує проблему нестачі зручних доменних імен, відкриваючи нові можливості для просування своїх товарів і послуг.
- Закріплює індивідуальність особи або компанії.

Багатомовний домен IDN - це звичайний домен латиницею, який містить обов'язковий префікс xn- (ознака IDN-домену) і закодовану частину, яка може бути перекодована в слово на національному алфавіті. Наприклад, в браузері можна з клавіатури набрати «*мійдомен.ua*» і його буде автоматично перекодовано в «*xn-d1ack1chcc.ua*».

На перший погляд зручність є очевидною: компанії отримують прості імена, що легко запам'ятовуються, типу «*сайт.ua*», які просто продиктувати клієнтові. Але це лише на перший погляд. Нажаль, користувачам доведеться стежити, в якій розкладці слід набирати першу частину домену, а в якій другу,

адже *SU*, *COM* і *NET* все одно доведеться набирати латиницею. А ще доведеться пам'ятати, що клієнту слід набрати саме «сайт», кирилицею, а не в звичний спосіб *site* латиницею.

Національні домени верхнього рівня NLLA

Ідею втілення повного IDN затверджено в 2010 році. Тоді з'явилися перші інтернаціоналізовані домени першого рівня: — *امارات*. — ОАЕ, *السعودية*. — Саудівська Аравія, *مصر*. — Єгипет, *.рф* — Росія.

В Україні делегування домену *.укр* відбулося в 2013 році, перший сайт в національній доменній зоні – *президент.укр*. Реєстрація в українському *.УКР* розпочалася в 2014 р.

Конвертація інтернаціоналізованих доменів Punycode

Для вирішення питання з підтримкою національних алфавітів для доменних імен існують спеціальні стандарти IDNA (Internationalizing Domain Names in Applications). Ці стандарти описують технології кодування символів, які не входять в таблицю ASCII в доменних іменах. Одним зі стандартів - RFC 3492 описується спеціальне перетворення символів - Punycode, що дозволяє конвертувати набір символів в кодуванні Unicode (містить всі національні алфавіти) в набір символів, що підтримує служба DNS (кодування ASCII).

Наприклад, IDN-домен, записаний з використанням кирилиці як «сайт.com», після Punycode-конвертації матиме вигляд «xn--80aswg.com».

Punycode-перетворення застосовується також і до зони (домену верхнього рівня - УКР). Домен Кириличний домен «сайт.укр» після Punycode-перетворення буде виглядати «xn--80aswg.xn--j1amh».

Для підтримки IDN-доменів досить, щоб браузер користувача підтримував Punycode-перетворення. На даний момент останні версії всіх популярних браузерів підтримують Punycode. У тому випадку, якщо браузер не підтримує Punycode, а також, якщо необхідно отримати ім'я домену в його «реальному поданні», можна скористатися спеціальними веб-сервісами виконують Punycode-перетворення введених користувачем доменних імен.

Punycode converter <https://www.punycoder.com/>

Нові домени верхнього рівня New gTLDs

Корпорація ICANN, яка керує адресним простором мережі Інтернет, затвердила порядок створення нових доменів верхнього рівня - програму New gTLD. Згідно правил претендувати на отримання власної доменної зони відтепер може будь-яка юридична особа.

Раніше існувало всього лише 22 доменів верхнього рівня (gTLD), включаючи відомі .com, .org і .net. На сьогодні за програмою New gTLD створено більше тисячі нових доменних зон. Серед них є домени, що збігаються з загальноживаними словами - .PHOTO, .BLOG, .SHOP, .DOG, а також зони, що відповідають назвам найбільших світових брендів - .CANON, .VOLKSWAGEN, .YANDEX та найменуванням територій - .AFRICA, .BERLIN, .VEGAS тощо.

Зареєструвати імена сайтів в нових доменних зонах користувачі зможуть в уповноваженого реєстратора, наприклад, https://www.ukrnames.com/reg_domains/new_gtld.jsp

Заявка на отримання New gTLDs є складним процесом, оскільки кандидат подає заявку на створення реєстру та підтримку системи доменних імен Інтернету. Це передбачає ряд важливих обов'язків, оскільки оператор New gTLDs керує частиною інфраструктури видимого Інтернету.

Подавати заявку на New gTLDs може будь існуюча організація державного або приватного сектора, що відповідає вимогам до кандидатів на нові домени. Кандидати повинні продемонструвати операційні, технічні та фінансові можливості керування реєстром і виконувати певні додаткові вимоги.

Збір за розгляд заявки становить \$ 185 000. При реєстрації кандидати мають внести заставу в розмірі \$ 5000 доларів за кожную заявку. Дану заставу буде враховано у сумі збору за розгляд заявки. Залежно від специфіки розгляду конкретної заявки можуть стягуватися й інші збори.

Які можливості отримує власник власного домену верхнього рівня?

- З доменом верхнього рівня великі компанії та організації отримують можливість формування єдиного адресного простору в Інтернеті для розміщення інформації про свої продукти, послуги, напрями діяльності. Домен верхнього рівня дозволяє створювати

адреси виду PRODUCT.YOURBRAND, USLUGA.YOURBRAND, ПРОДУКТ.ВАШБРЕНД, УСЛУГА.ВАШАКОМПАНІЯ, які у користувачів однозначно будуть асоціюватися з певною компанією або організацією.

- Міста та географічні регіони можуть використовувати домени верхнього рівня, що збігаються з їх назвами, для розвитку свого туристичного потенціалу, підвищення впізнаваності в Інтернеті.

- Домени верхнього рівня можуть служити для ідентифікації різних спільнот в Інтернеті. Наприклад, подібну функцію успішно виконує домен tattoo, який об'єднує в віртуальному просторі спільноту любителів тату.

- В якості домену верхнього рівня може бути зареєстровано будь-яке загальноживане слово - наприклад, .ТАХІ, .БЛОГ. Подальша платна реєстрація доменних імен в таких доменах може стати прибутковим напрямом бізнесу.

Із збільшенням числа доменних імен верхнього рівня з'являється більше можливостей для залучення цільової аудиторії у громадських, некомерційних, неурядових організацій і фондів.

Службові та зареєстровані домени

Згідно з документом RFC 2606 зарезервовано домени верхнього рівня:

- .example - для прикладів.
- .invalid - для прикладів неробочих адрес.
- .localhost - для уникнення конфліктів з традиційним використанням localhost.
- .test - для використання в тестах.
- .arpa - використовується виключно для інфраструктури Інтернету.

До зарезервованих імен другого рівня належать:

- example.com, example.org і example.net, example.edu

Зарезервовані імена другого рівня зроблено для використання в посібниках та прикладах налаштування програмного забезпечення без ризику виникнення конфліктів з вже зареєстрованими доменами або доменами, які можуть бути зареєстровані в майбутньому. Таким чином, при написанні документації можна бути впевненим, що не буде нанесено шкоди

кому-небудь, якщо користувач спробує використати приклад без зміни.

Якщо набрати ці адреси, то користувач побачить повідомлення:

Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

2. Вибір доменного імені

Перше, з чим знайомиться потенційний відвідувач ще до відвідування сайту – це доменне ім'я. Нижче наведено чинники, які потрібно враховувати при виборі доменного імені сайту, що впливають як на сприйняття людиною, так і на сприйняття пошуковими системами.

Стислість

За можливістю варто вибирати коротке доменне ім'я з кількох причин. Під нього легше зробити логотипи, будувати брендинг, воно швидше впізнається, менше можливостей неправильно його написати. Деякі компанії підбирають доменні імена, що містять понад 50 символів, проте, за їх словами, таким чином можна повністю викласти ім'я компанії. Втім, так чинити нерозумно. Довге доменне ім'я не поміщається в стандартних формах, в рекламних оголошеннях або в програмах просування пошукових систем. Слід прагнути робити доменне ім'я відносно коротшим.

Не варто зловживати аббревіатурами. Наприклад, `uwf.kiev.ua` - належить Українському Жіночому Фонду, `unba.lviv.ua` – Рада адвокатів міста Львова. Такі доменні імена пересічний користувач навряд чи зможе зрозуміти. Втім, часто кілька букв, наводить на однойменну назву бренду. Наприклад: домен `stb.ua` (Телеканал СТБ) чи `aif.ua` (Аргументи і Факти в Україні).

Запам'ятовуваність

Влучне доменне ім'я не забувається. Як правило, добре запам'ятовуються загальні імена, наприклад: `apteka.ua`, `football.ua`, `price.ua`. Добре виглядають імена, що складаються з кількох змістовних слів: `superjob.ua`, `neogame.com.ua`, `freetime.com.ua`. Існує й група унікальних імен, наприклад Amazon, Google, Yandex, які

вже самі за себе говорять. При підборі доменного імені, можна спробувати утворити риму або пограти з повторенням звуків, наприклад, HostPro, Google або Okko. Комбінація звуків та слів в кінцевому підсумку, може створити хороше ім'я.

Змістовність

Найкраще прив'язувати доменне ім'я до тематики сайту, щоб за доменним іменем можна було здогадатися про тему сайту: drummer.org.ua (сайт українських барабанщиків), prikols.dp.ua (дніпропетровський гумористичний портал).

Але трапляються сайти, доменні імена яких, не збігаються з їх тематичним вмістом: lvtech.com.ua (Львівська фабрика більярдів), csn.lpnu.ua (Науковий журнал «Комп'ютерні системи та мережі»). Зрозуміло, що вся справа в незвичайності, але це вносить дисонанс у сприйнятті.

Часто власники використовують власні імена та ніки для імені домену. З цим варто бути обережним. Якщо сайт буде використовуватися в якості блогу, тоді доменне ім'я-ник цілком підійде. Якщо ж власник домену хоче представити свої послуги чи роботи, то варто в доменному імені позначити напрямок. Наприклад, за доменом kellyanndesign.net відразу видно, що сайт присвячений дизайну від Kelly Ann.

Чіткість

Вдале підібране доменне ім'я важко переплутати з іншими. Немає сенсу шукати доменне ім'я, співзвучне з чийось, що написано через дефіс або до якого дописано кілька букв. Виникне проблема непорозуміння. Наприклад, деякі сайти виглядають майже однаково, різниця тільки в деяких символах, наприклад, kompass.ua і compass.ua; apostrof.ru і apostroff.ru; lovemy.com.ua і loveme.com.ua.

Законодавство про інтелектуальну власність та захисту торгових марок запобігає введенню в обману покупців. Якщо власник схожого доменного імені першим зареєстрував свою комбінацію символів, то для подібного домену з'являється серйозна загроза, принаймні, для використання такого імені як брэнда. Перед реєстрацією слід перевірити базу Укрпатенту - для реєстрації торговельної марки в Україні (довідкові послуги такого типу - платні) або базу країни, в якій зареєструється торговельна марка.

Пильної уваги потребує транслітерація - і написання кирилиці латинськими символами. Є кілька варіантів написання букв: х (h, kh), ц (ts, z), в (w, v), до (c, k, ck), ф (f, ph). Наприклад, доменні імена foto.ua і photo.ua звучать абсолютно однаково – фото, але написання літер є різним.

Просте написання

Чим довше і важче доменне ім'я - тим важче буде відвідувачам правильно його написати. Коректне введення довгого доменного імені може виявитися занадто складним. Наприклад, webcruftstuff.net, або багаторазовий повтор складів як blablablablablaba.com, в якому кількість повторень запам'ятати просто нереально.

Є інші випадки неправильного написання доменного імені. Наприклад, для компанії МКС логічно було купити домен mks.ua, при цьому, домен mks.ua теж дуже добре б підійшов даної компанії. Враховуючи особливості українських користувачів мережі, все-таки можуть знайтися бажаючі використовувати домен mks.ua, щоб подивитися сайт компанії МКС.

Другий приклад: компанія Меркс (Merx) може скористатися доменними іменами merx.ua, merx.com.ua, merks.com.ua, merks.ua.

Наочність

Ідеальна ситуація, коли ім'я домену збігається з назвою компанії. Якщо бажане ім'я вже зайнято, варто підібрати функціональне ім'я, яке виражає напрямок послуг компанії.

З моменту візиту першого відвідувача - домен автоматично стає брендом, обличчям, візиткою сайту. Саме за доменом ресурс будуть ідентифікувати в мережі. Краще підбирати ім'я для компанії та домену одночасно. На цьому етапі, вдале ім'я підібрати досить просто. Так, маючи домен elt109.com.ua, важко буде підібрати назву для компанії, враховуючи і прив'язку до роду діяльності.

Для уточнення роду діяльності можна використовувати різні зони. Так, некомерційні організації часто реєструють сайти в зоні .org, комерційні - .com, інформаційні - .info, урядові - .gov. Наприклад, сайт про музику в Україні має доменне ім'я music.in.ua., а різниця у вмісті сайтів rada.com.ua і rada.gov.ua очевидна виходячи вже з розширень.

Цільова спрямованість

Гарне доменне ім'я - звучить переконливо для цільової аудиторії. За можливості, варто отримати домен першого рівня; найпоширеніші: .com, .net, .org; для України - .ua, .com.ua (третій рівень). Можна спробувати також і .biz, .info, або .cc, .ws, .tv, і .to. При виборі імені потрібно враховувати цільову спрямованість і географічний обхват аудиторії. Якщо сайт орієнтовано на українського відвідувача - то найкраще отримати домен в зоні .ua. Якщо планується бізнес світового масштабу - то користувачі схильні в більшості використовувати розширення .com або .biz. Доменне ім'я повинне бути таким, яким очікують бачити його люди. Для України це .ua, .com.ua; рідше .net.ua, .org.ua, biz.ua.

Покупка потрібного домену у власника

При підборі доменного імені часто виникає ситуація, що всі цікаві, красиві імена вже зайняті. Переглядаючи варіанти, з'ясовується, що більша їх частина до сайтів не прив'язана, а просто зайнята для подальших маніпуляцій. Якщо компанія серйозно націлена саме на певне ім'я, то можна спробувати купити його у власника.

Спочатку слід з'ясувати, який сайт знаходиться за даною адресою, ретельно його переглянути та проаналізувати. Покинутий сайт з низькою відвідуваністю і рідкісними оновленнями, швидше за все, не потрібен своєму власникові і він швидше від нього позбавиться і продасть домен. Контакти власника, зазвичай, розміщуються на самому сайті або на сервісі Whois. Якщо сайт належить приватній особі, то особисті дані будуть приховані, але контактна інформація: телефон, електронна адреса - доступні завжди.

Ціни

Вже існуючий готовий домен купити можна, але він коштує значно дорожче, ніж новий - в десятки, а іноді сотні разів. Вартість доменних імен не є дуже дорогими, а щорічна оплата виносить приблизно 20-50\$, тому нинішній власник домену краще раз на рік продовжить реєстрацію, ніж продасть його. Домени з давно створеними сайтами вже мають певну вагу для пошукових систем, кількість посилань і своїх користувачів. Ці показники є основними критеріями для підвищення вартості домену.

Існують компанії, які реєструють доменні імена і проводять аукціони вивільнених доменів. Це імена, реєстрацію на які не було продовжено вчасно. Якщо з певних причин власник забув продовжити реєстрацію, то ім'я домену блокується на один місяць. По закінченні відведеного терміну, воно знов стає загальнодоступним. Компанії-реєстратори скуповують звільнені імена і виставляють на продаж. Їх вартість значно перевищує вартість нового імені.

Вартість покупки

Супердорогі домени продаються рідко. Висока вартість оригінального домену на аукціоні припускає терпіння і завзятість, оскільки необхідно певний час його притримувати, вичікуючи зручний момент. З ростом популярності Інтернету зростають ціни на доменні імена.

Вартість доменів розраховується виходячи з купівельного попиту. Інтереси до доменів, які мають географічні розширення країн, що швидко розвиваються, безсумнівно виростуть в ціні разом із зростанням купівельного попиту і економіки країн. Просунуті домейнери знають про це і заздалегідь інвестують в покупку доменів.

Адекватна ціна звільненого домену залежить від кількох чинників:

- **Довжина і запам'ятовуваність.** Чим коротше домен, тим легше його запам'ятати, і відповідно тим дорожче вартість.
- **Вік домену.** Чим раніше його зареєстрували, тим він дорожчий.
- **Історія.** Не завжди вік буде відігравати важливу роль, тому, якщо домен зареєстрували 10 років тому, і за цей час кілька разів змінювався контент або тематика сайту, або ж він взагалі був у бані пошукових систем (через нечесні методи просування), то цей фактор істотно знизить вартість. Перед покупкою вивільнених доменних імен, обов'язково потрібно провести перевірку на наявність бану. Якщо цей момент було упущено, в такому випадку існує можливість «врятувати» домен, але для цього потрібно витратити багато часу і зусиль, щоб довести корисність ресурсу.
- **Тематика.** Цей параметр не має великого впливу на ціну, оскільки людина, яка хоче створити сайт про їжу, спочатку буде вибирати собі домени, в назвах яких зустрічається ця тематика.

- **Показники авторитетності домену PageRank (Google).** Ці показники мають велике значення для оцінки вартості домену, особливо, якщо покупець має мету продавати з купленого сайту. У підсумку, чим вище ці показники, тим дорожче домен.

- **Наявність в каталозі DMOZ.** Якщо сайт додано до DMOZ-каталогу, це враховується пошуковими системами при визначенні рівня довіри до нього. Якщо сайт є в цьому каталозі, то це з великою ймовірністю говорить про хорошу історію домену (відсутність банів і штрафів від пошукових систем).

- **Зона домену** має різний вплив на ціну, і це залежить від ситуації. Наприклад: домен *likar.ua* апріорі коштує дорожче, ніж *likar.lviv.ua*, оскільки перший охоплює аудиторію країни, а другий - тільки міста. Втім, якщо знайдеться фірма, якій потрібен такий домен саме для роботи у Львові, то вона заплатить за це чималі гроші.

- **Наявність дефіса або цифр у назві** може зменшити ціну на домен.

- **Невеликий вибір.** Якщо ідентичне доменне ім'я вже зайнято в інших зонах, то це може збільшити ціну. Наприклад, якщо є вільний домен *kino.com.ua*, то домен *kino.biz.ua* буде менш привабливий для покупця і, відповідно, оцінюватися значно менше.

Омонімічні домени

Омонімічні домени - домени верхнього рівня, найчастіше національні, що за написанням або звучанням схожі з різними словами. Зазвичай, належать невеликим маловідомим державам. Сайти в таких доменах реєструють лише через співзвуччя, а не виходячи з їхнього справжнього розташування.

Трапляються випадки, коли приватна компанія купує право реєстратора в національному домені лише через його оригінальне звучання. Саме це сталося з доменом тихоокеанської держави Тувалу (.tv).

.am - національний домен Вірменії. Збігається з діапазоном радіостанцій АМ і з часом «до полудня».

.cd - національний домен Демократичної Республіки Конго. Збігається зі скороченням для компакт-диска.

.dj - національний домен Джибуті. Збігається зі скороченням «діджей».

.fm - національний домен Федеральних Штатів Мікронезії. Збігається з діапазоном радіостанцій FM. Приклад: last.fm.

.im - національний домен Острову Мен. Збігається зі скороченням instant messaging («миттєві повідомлення»).

.in - національний домен Індії. З англійської мови перекладається як прийменник «в».

.io - національний домен Британської території в Індійському океані. Збігається зі скороченням Input Output.

.is - національний домен Ісландії. Збігається зі словом is, формою третьої особи однини англійського дієслова to be. Приклад: who.is

.it - національний домен Італії. Збігається зі скороченням ІТ («інформаційні технології»), а також з англійським займенником it. Приклад: ok.undo.it

.li - національний домен Ліхтенштейну. Збігається із закінченням дієслів і іменників у російській мові. Приклади: zadolba.li, zastuka.li.

.md - національний домен Молдови. Збігається зі скороченням MiniDisc, зі скороченням англ. must die («повинен померти») і з скороченням лат. Medicinae Doctor («доктор медичних наук»).

.me - національний домен Чорногорії. Збігається з займенником «мене», «мені» в англійській та інших європейських мовах.

.ms - національний домен Малих Антильських островів. Збігається зі скороченням назви компанії Microsoft.

.net - загальний домен верхнього рівня. Співзвучний з російським словом «ні». Приклади: mozga.net, lishnih.net.

.nu - національний домен Ніуе. Співзвучний зі словом «ню».

.pm - національний домен Сен-П'єр і Мікелон. Збігається з часом «після полудня».

.tm - національний домен Туркменії. Збігається з позначенням «ТМ» («товарний знак»).

.tv - національний домен Тувалу. Збігається зі скороченням слова television («телебачення»).

.ws - національний домен Самоа. Збігається зі скороченням web site («веб-сайт»).

Domain Hack, Dnhack, Домен-Хак

Domain Hack - це використання доменного імені та зони включно, з метою створення такої адреси, яка би представляла з себе звучне слово, речення, словосполучення, поняття і т.п.

Domain Hack - це підбір імені домену таким чином, щоб позначення доменної зони складало частину назви сайту (наприклад, del.icio.us, neku.ru, wua.la). Причому, в даному випадку слово hack не повинно асоціюватися з хакерством або порушенням закону, оскільки domain hack зовсім не порушує законів. Domain Hack можна сміливо використовувати в своїх доменах, це виглядає ексклюзивно і креативно.

Спробувати створити цікаве ім'я можна за допомогою сервісу Domain Hacks (<http://xona.com/domainhacks/>). Там же опубліковані найцікавіші хаки - inter.net, hi.fi, tetr.is, movi.es, windo.ws, blo.gs, sear.ch, doma.in.

Домен з type-in фактором

Щоб навмисно неправильно написана назва сприймалася як справжній домен можна застосувати type-in фактор, наприклад:

- пропуск літери (наприклад, gogle.ua);
- заміна літери (наприклад, wideo.ua, prezedent.ua, plase.ua, mtc.ua), можливі варіанти: v <-> w, k <-> c, c <-> s, y <-> i;
- помилка в послідовності букв (наприклад, spotr.ua, soprt.ua, ulnimit.ua), можливі варіанти: rt, er, nm, as, op;
- невірне написання слова (наприклад, adversiting.ua);
- невірне застосування кількості букв (наприклад, odnoklasniki.ua);
- пропуск точки після "www" (наприклад wwvdomainlist.ua);
- спрямоване написання (наприклад, isquins.ua);
- помилка в послідовності слів (наприклад, whoisdomain.ua);
- заміна букви двома буквами (наприклад, експерт.ua, exspert.ua, gugle.ua), можливі варіанти: x <-> ks, x <-> xs, j <-> zh, j <-> ya, u <-> oo;
- помилка написання букви (наприклад, qip.ua <-> qiq.ua);
- так звана "олбанська мова" (наприклад, odnoklassneg.ua);
- позбавлення від тире (наприклад, whoissearch.ua);

- написання з множини в однині (наприклад, film.ua);
- неправильна розкладка: meta.ua -> ьуеф (.com) (браузер Opera автоматично додає .com при натисканні кнопки "Enter" і переходить на сайт, теж саме відбудеться і в Mozilla, але натискати треба CTRL + Enter).

Домен з Type-in фактором буде правильним, якщо для створення назви застосовано один з наведених підходів, гірше - два відразу, зовсім погано - три відразу. Якщо домен підходить під чотири і більше умов відразу, то це вже буде не type-домен, а зовсім інший домен.

Практика показує, що до 20% відвідувачів, що вводять назву в адресне поле браузера допускають помилку у написанні назви і попадають на добре підібраний. В теорії до 25% відвідувань вихідного правильного домену йдуть на правильно підібраний домен з Type-in фактором. Відсоток залежить від складності домену, обраного type-фактора і може сильно змінюватися.

Кіберсквотинг

Кіберсквотинг (Cybersquatting), «захоплення доменів» - це реєстрація доменних імен, що містять торгівельну марку, яка належить іншій особі з метою їх подальшого перепродажу або несумлінного використання. Людей, що практикують такі дії, називають кіберсквотерами.

Захоплення доменів і торгових марок

Відповідно до законодавства, товарний знак має пріоритет перед доменним іменем, тобто власник товарного знака може заборонити використання свого знаку в доменному імені. Фактично це означає можливість відібрати домен у кіберсквотера. Звідси виникає можливість «зворотного захоплення» домену, тобто здійснення операції відбору домену через реєстрацію відповідного товарного знака. Проте, знаючи про таку можливість, багато власників дорогих доменних імен реєструють відповідні їм товарні знаки.

Кіберсквотинг в Україні

Явище кіберсквотингу притаманне і для України, правда не в тих обсягах, що й у світі. Наприклад, за адресою <https://ua.auctions.godaddy.com/> виведено перелік доменів, які можна придбати на аукціоні.

Інформацію про те, скільки в Україні таких «контор» і який обсяг зареєстрованих доменів припадає на них точно не відомо. Теоретично можна робити певний аналіз, вибирати популярні імена і назви суб'єктів, що займають ті чи інші адреси. На практиці це є доволі складно, бо, наприклад, на одну корпорацію типу Suzuki або LLC може припадати сотні зареєстрованих доменів.

Кіберсквотинг в Україні відбувається практично у всіх доменних зонах. Враховуючи, що домени коштують недорого, їх завжди можна продати дорожче, ніж зареєструвати. Так, за звучний домен, зареєстрований на \$ 10, можна виручити в середньому близько \$ 150-300. У той же час в домені .com.ua зустрічаються імена, вартість яких сягає кількох тисяч доларів.

Тобто, про захмарні суми, які зустрічаються у світі, в Україні поки не йдеться. У той же час західні бренди, навчені гірким досвідом доменних воєн, готові витратити великі гроші на відстоювання свого домену.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Зав'ялець Ю.А. Web-технології та web-дизайн : конспект лекцій. Чернівці, 2014. 90 с
2. Копішинська О. П. Методичні рекомендації для самостійної роботи з дисципліни «Вебдизайн і розробка клієнтської частини веб-застосування». Полтава : ПДАА, 2020. 28 с. URL : <http://dspace.pdaa.edu.ua:8080/handle/123456789/9745>
3. Молчанов В. П. Основи проектування WEB-видань : конспект лекцій. Харків : Вид. ХНЕУ, 2008. 168 с.
4. Молчанов В. П. Технології WEB-дизайну : конспект лекцій. Харків : Вид. ХНЕУ, 2011. 212 с.
5. Молчанов В. П., Андрющенко Т. Ю. Методичні рекомендації по виконанню лабораторних робіт з навчальної дисципліни "Основи проектування WEB-видань" для студентів спеціалізації "Комп'ютеризовані технології та системи видавничо-поліграфічних виробництв" усіх форм навчання. Харків : Вид. ХНЕУ, 2009. 84 с.
6. Пасічник В. В., Пасічник О.В., Угрін Д.І. Веб-технології та Веб-дизайн : підручник. Львів : «Магнолія 2006», 2018. 336 с.
7. Погребняк Б. І., Булаєнко М. В. Web-технології та Web-дизайн : конспект лекцій (для студентів денної та заочної форм навчання спеціальностей 122 – Комп'ютерні науки, 126 – Інформаційні системи та технології та 151 – Автоматизація та комп'ютерно-інтегровані технології). Харків. нац. ун-т гір. госп-ва ім. А. Н. Бекетова. Харків : ХНУГГ ім. А. Н. Бекетова, 2018. 19 с.
8. Спирінцев В.В., Гнатушенко В.В., Волковський О.С. Web-технології та Web-дизайн : HTML, CSS : навч. посіб. Дніпро : «Ліра», 2017. 163с.

Навчальне видання

Веб-технології та веб-дизайн

Конспект лекцій

Укладачі:

Шебаніна Олена В'ячеславівна
Тищенко Світлана Іванівна
Хилько Іван Іванович
Крайній Володимир Олексійович
Пархоменко Олександр Юрійович
Чуриков Андрій Анатолійович

Формат 60x84 1/16. Ум. друк. арк. 6.
Наклад 50 прим. Зам. № _____

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від
20.02.2013 р.