

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ПОЛІТЕХНІКА»



МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних занять
з дисципліни «Веб-технології та веб-дизайн»
для студентів 2 курсу
спеціальності 022 Дизайн
освітня програма – Архітектурний дизайн

Одеса
ОНПУ
2024

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ПОЛІТЕХНІКА»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних занять
з дисципліни «Веб-технології та веб-дизайн»
для студентів 2 курсу
спеціальності 022 Дизайн
освітня програма – Архітектурний дизайн

Затверджено
на засіданні кафедри ІТПД
Протокол №1 від 28.08.2024 р.

Одеса
ОНПУ
2024

Методичні вказівки до лабораторних занять з дисципліни «Веб-технології та веб-дизайн» для студентів 2 курсу першого (бакалаврського) рівня вищої освіти, що навчаються за спеціальністю 022 – Дизайн, освітньо-професійна програма – Архітектурний дизайн / Укл.: О.В. Рибак, Д.М. Солодкий. – Одеса: ОНПУ, 2024. – 49 с.

Укладач: Рибак О.В., к.т.н., доцент
Солодкий Д.М., асистент

Методичні вказівки містять завдання до лабораторних робіт з дисципліни «Веб-технології та веб-дизайн», передбачені навчальною програмою протягом 4 семестру підготовки бакалаврів, які навчаються за спеціальністю 022 – Дизайн напрямку підготовки Архітектурний дизайн. Викладено перелік лабораторних робіт відповідно до навчальної програми, мету роботи, основні теоретичні положення, зміст та завдання до роботи і послідовність їх виконання. Також наведено рекомендації щодо оформлення звіту з лабораторної роботи та контрольні запитання для самоперевірки до кожної теми. Лабораторні заняття покликані сформувати вміння та навички практичного застосування набутих знань шляхом виконання здобувачами прикладних завдань відповідно до теми роботи. Методичні вказівки призначаються для здобувачів денної та заочної форми навчання.

ЗМІСТ

	стор
ВСТУП	4
ЛАБОРАТОРНА РОБОТА 1.	5
ЛАБОРАТОРНА РОБОТА 2.	7
ЛАБОРАТОРНА РОБОТА 3.	9
ЛАБОРАТОРНА РОБОТА 4.	11
ЛАБОРАТОРНА РОБОТА 5.	13
ЛАБОРАТОРНА РОБОТА 6.	15
ЛАБОРАТОРНА РОБОТА 7.	18
ЛАБОРАТОРНА РОБОТА 8.	21
ЛАБОРАТОРНА РОБОТА 9.	24
ЛАБОРАТОРНА РОБОТА 10.	26
ЛАБОРАТОРНА РОБОТА 11.	28
ЛАБОРАТОРНА РОБОТА 12.	31
ЛАБОРАТОРНА РОБОТА 13.	35
ЛАБОРАТОРНА РОБОТА 14.	41
ЛАБОРАТОРНА РОБОТА 15.	45
СПИСОК ЛІТЕРАТУРИ	49

ВСТУП

Методичні вказівки до лабораторних занять з дисципліни «Веб-технології та веб-дизайн» містять перелік лабораторних робіт, які передбачені навчальною програмою до вивчення на 2 курсі протягом 4 семестру підготовки бакалаврів, що навчаються за спеціальністю 022 – Дизайн напрямку підготовки Архітектурний дизайн.

Лабораторні заняття покликані сформувати вміння та навички практичного застосування набутих знань шляхом виконання здобувачами прикладних завдань. *Основна мета* лабораторних занять полягає у розширенні, поглибленні й деталізації знань, отриманих на лекціях та в процесі самостійної роботи, і підвищенні рівня засвоєння навчального матеріалу, а також закріплення знань, умінь і навичок вирішення практичних задач.

В процесі вивчення дисципліни «Веб-технології та веб-дизайн» здобувачі опановують сучасні можливості інформаційних технологій, інструменти та шляхи розробки користувацьких веб-інтерфейсів та дизайну веб-сайтів. Виконання лабораторних робіт, представлених у даних методичних вказівках, забезпечить формування у здобувачів умінь та навичок веб-дизайну та розробки веб-сторінок, а також вміння застосовувати сучасні методики роботи та технічної підтримки веб-сайтів для подальшої професійної діяльності.

Перед виконанням кожної лабораторної роботи студенти зобов'язані ознайомитися з основними теоретичними положеннями, порядком проведення роботи та відповідними положеннями охорони праці при проведенні лабораторних робіт.

Звіт з лабораторної роботи складається під час підготовки студента до виконання роботи і в процесі її виконання. Звіт повинен включати: порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Якщо це зазначено у завданні до роботи, необхідно також представити коди розроблених веб-сторінок та створених файлів зі стилями. Кожен звіт завершується висновком до виконаної лабораторної роботи.

Після виконання здобувач захищає лабораторну роботу на одному з наступних занять. При цьому знання відповідей на контрольні запитання до лабораторної роботи є обов'язковим.

ЛАБОРАТОРНА РОБОТА № 1.

Тема: Знайомство з принципами роботи пошукових систем і браузерів.

Мета роботи: Навчитися розрізняти браузери і пошукові системи, розуміти принципи їхньої роботи.

Короткі теоретичні відомості

Пошукова система – це програмний комплекс, що надає користувачам можливість доступу до необхідної інформації в інтернеті за допомогою пошуку в універсальному сховищі даних про веб-сайти, проіндексовані зазначеною пошуковою системою.

Браузер – це програма на комп'ютері, за допомогою якої можна переглядати власні веб-сторінки, результати видачі пошукових систем, сторінки різних сайтів в інтернеті, переходити між ними.

Ключові відмінності:

1. Неможливо отримати доступ до будь-якої інформації в інтернеті без веб-браузера. Знаючи адресу сайту, можна перейти на нього, не використовуючи пошукову систему.

2. Якщо користувач не має конкретної URL-адреси веб-сторінки, за допомогою пошукової системи можна знайти необхідну інформацію за фразою, словом або словосполученням, введеним у пошуковий рядок.

3. Браузер використовується для пошуку конкретного веб-сайту, а пошукова система – для сортування та фільтрації відповідної до запиту інформації.

4. Браузер працює як незалежна програма, натомість пошукова система не може працювати без браузера.

5. Для подальшої роботи, браузер необхідно встановити на комп'ютері або іншому пристрої. Користування пошуковою системою не потребує встановлення програмного забезпечення та передбачає доступ через браузер.

Формуючи запит до пошукової системи, кожен користувач має інформаційну, навігаційну або транзакційну мету. Виходячи з цього, виділяють чотири цільових типи запитів:

1. Інформаційні запити, коли користувача цікавить конкретна інформація, але для нього не принципово, на якому ресурсі він її отримає. Наприклад, «розклад потягів Київ-Одеса», «як приготувати салат з авокадо», «огляд фільму Гаррі Поттер і філософський камінь».
2. Навігаційні запити здійснюються, коли користувач хоче знайти конкретний, визначений сайт, вводячи в пошуковий рядок назву веб-сайту або організації. Наприклад, «сайт ощадбанка», «Укрзалізниця», «вікіпедія», «facebook».
3. Транзакційні запити передбачають виконання користувачем певної дії – купити, завантажити, замовити, подзвонити. Прикладами таких запитів є: «купити квіти з доставкою», «завантажити відеогру», «zareestruvati poshtu gmail».
4. Нечіткі запити, коли наміри користувача не закладені в запит і залишаються для пошукової системи нез'ясованими. Прикладами таких запитів можуть бути: «комп'ютер», «фільми», «квіти».

В залежності від типу пошукового запиту, пошукова система визначає релевантність того чи іншого сайту і його місце в списку результатів пошукової видачі.

Завдання: Оглянути різні види браузерів і пошукових систем. Побудувати логічну схему обробки запиту. Визначити типи пошукових запитів.

Хід роботи

1. Визначте, які браузери встановлені на вашому комп'ютері та смартфоні.
2. Перелічіть відомі вам пошукові системи та браузери.
3. Виконайте пошук сайту за відомою вам URL-адресою.
4. Виконайте пошук сайтів за фразою, введеною в рядок пошукової системи.
5. Побудуйте логічну схему обробки запиту пошуковою системою у відповідності до теоретичного матеріалу, розглянутого на лекції.
6. Визначте типи пошукових запитів, розглянутих протягом лабораторного заняття.
7. Наведіть власні приклади пошукових запитів різних типів.
8. Зробіть висновок по роботі.

Домашнє завдання:

Опишіть, у чому полягає відмінність пошукової системи від браузера. Визначте, якою пошуковою системою та браузером ви зазвичай користуєтесь. Підготуйте огляд, які пошукові системи набули поширення у різних країнах світу.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Дайте визначення поняттям «веб-сайт», «браузер», «пошукова система».
2. Подумайте, яка пошукова система є найпопулярнішою в усьому світі.
3. Поясніть різницю між цільовими типами пошукових запитів.
4. Опишіть принцип роботи пошукових систем.
5. Визначте, що собою являє процес індексації сайту.
6. Перерахуйте чинники, що впливають на релевантність веб-сторінки до запиту користувача.

Література: [1] с. 7–12, 32–34; [3] с. 28–36.

ЛАБОРАТОРНА РОБОТА № 2.

Тема: Вивчення особливостей функціонування мови пошукових запитів.

Мета роботи: Навчитися використовувати основні можливості інформаційно-пошукової мови.

Короткі теоретичні відомості

Мова, якою сформульовані запити до пошукових систем, називається інформаційно-пошуковою мовою, або мовою пошукових запитів. Інформаційно-пошукова мова складається з логічних операторів, морфології мови, регістра слів, префіксів обов'язковості, можливості враховувати відстань між словами і розширеного пошуку. Подібне представлення запиту допомагає швидше орієнтуватися в індексних базах.

Набір команд мови запитів може змінюватися в залежності від особливостей конкретної пошукової системи. Але існують загальні правила, які застосовують до усіх запитів. Розглянемо найбільш розповсюджені з них.

- Команди логічного об'єднання і виключення в запиті дозволяють додавати будь-які слова в текст або виключати їх з тексту. Слово, відмічене «+», обов'язково буде присутнім в документах, які знайде пошукова система за запитом. Слова, відміченого «-», у результатах видачі не буде. Команди логічного об'єднання і виключення повинні бути написані разом зі словом, до якого вони відносяться. У протилежному випадку пошукова система почне розглядати їх як елементи запиту, а не як команди.
- Пошук за точною фразою дозволяє перерахувати слова, які обов'язково повинні зустрічатися в межах одного речення в заданій формі.
- Об'єднання запитів за допомогою логічного оператора АБО (позначається «OR» в Google) дає можливість здійснювати пошук по веб-сторінках, в яких присутнє тільки одне з перерахованих слів.
- Пошук по визначеному сайту здійснюється шляхом запису слова «site:» перед адресою сайту або доменом.
- Пошук сайтів зі схожим контентом здійснюється шляхом додавання слова «related:» перед зазначеною веб-адресою.
- Перегляд результатів пошуку всередині числового діапазону здійснюється шляхом введення двох крапок «..» між двома числами.
- Використання хештегів # (від англ. hashtag, де hash означає «знак решітки», а tag – це мітка) дозволяє групувати схожі повідомлення, які належать до певної теми або каналу у соціальних мережах.

Оператори мови пошукових запитів слід використовувати для отримання більш точних результатів. Пошукові системи таким чином можуть відсіяти непотрібні результати і показати найбільш релевантну інформацію, тобто ті веб-сторінки, які відповідають потребам користувача.

Завдання: Вивчити особливості використання різних операторів інформаційно-пошукової мови. Навчитися застосовувати команди логічного об'єднання і виключення. Виконати пошук відповідності за точною фразою.

Хід роботи

1. Спробуйте різні приклади використання операторів мови запитів:
 - команди логічного об'єднання і виключення;
 - об'єднання запитів за допомогою логічного оператора АБО;
 - пошук по визначеному сайту;
 - пошук сайтів зі схожим контентом;
 - пошук всередині числового діапазону.
2. Перейдіть на довільний сайт, скопіюйте будь-яку фразу з тексту на веб-сторінці, введіть її в пошуковий рядок Google у лапках та перевірте, чи дублює хтось інформацію з цього сайту.
3. Виконайте пошук груп повідомлень за хештегами компаній, тем, заходів, челенджів та туристичних пам'яток.
4. Зробіть висновок по роботі.

Домашнє завдання:

Визначте, коли слід застосовувати оператори інформаційно-пошукової мови. Опишіть особливості використання хештегів у різних соціальних мережах. Спробуйте створити власний хештег і подумайте, де його можна розмістити.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Подумайте, чи враховує пошукова система Google знаки пунктуації, які не відносяться до операторів мови запитів.
2. Визначте, як зміняться результати пошуку, якщо між спеціальним словом або символом та пошуковим запитом додати пробіл.
3. Перелічіть відомі вам оператори інформаційно-пошукової мови.
4. Дайте визначення поняттю хештег.
5. Наведіть приклади ситуацій, коли потрібно використовувати команди логічного об'єднання і виключення.

6. Опишіть, яким чином здійснюється пошук в інтернеті за точною фразою.

Література: [1] с. 15–17; [3] с. 48–53.

ЛАБОРАТОРНА РОБОТА № 3.

Тема: Створення простої веб-сторінки та її форматування, створення списків.

Мета роботи: Вміти створювати прості веб-сторінки та надавати їм певного вигляду, використовуючи для цього основні команди мови HTML.

Короткі теоретичні відомості

Мова HTML (HyperText Markup Language) – це мова розмітки документа, що описує форму відображення інформації на веб-сторінках у вікні веб-браузера.

Коди мови HTML, за допомогою яких проводять розмітку вихідного тексту, називають тегами. Під час відображення документа в браузері самих тегів не видно, але вони впливають на зовнішній вигляд документа. Виділяють парні – відкриваючий і закриваючий, та одиночні теги. Відкриваючі теги можуть містити атрибути, які впливають на ефект, що створюється тегом. Атрибути – це додаткові ключові слова, що відділяються від основного ключового слова тега і один від одного символами пробілу.

Для створення веб-сторінки необхідно використовувати теги: <html>, <head>, <title>, <body>. Форматування тексту здійснюється за допомогою тегу , а також його атрибутів size – для визначення розміру тексту та face – для задання типу шрифту.

Для виділення в тексті на веб-сторінці окремих параграфів необхідно скористатися тегом <p>. Вирівнювання параграфів, як і інших елементів веб-сторінки, по лівому краю, по правому краю, по центру або по ширині здійснюється шляхом додавання потрібного значення до атрибуту align.

Виділення заголовків на веб-сторінці відбувається за допомогою парних тегів <h1>, <h2>, <h3>, <h4>, <h5> та <h6>. Для задання накреслення курсивом використовується тег <i>, для напівжирного тексту – тег , для підкреслення – тег <u>. Верхній індекс задається парними тегами , а нижній індекс – тегами .

Існує 3 види списків, які можна розмістити на веб-сторінці: неупорядкований (маркований) список, упорядкований список, який ще називають нумерованим, а також список означень. Для створення неупорядкованого списку використовується тег , і кожен пункт списку виділяється тегом . Впорядкований список, пункти якого виділяються арабськими чи латинськими цифрами або літерами, створюється за допомогою тегу . Як і для маркованого списку, кожен пункт нумерованого списку виділяється тегом .

Списки означень складаються з двох частин: терміну та його визначення. Оголошується такий список тегом <dl>. Для вставки терміну застосовується тег <dt>, для вставки визначення – тег <dd>. Такий вид списків на веб-сторінках має вигляд словника і є найменш поширеним.

Завдання: Створити найпростіший html-файл довільного змісту. Застосувати теги форматування тексту. Продемонструвати приклади розміщення на веб-сторінці нумерованих та маркованих списків.

Хід роботи

1. Відкрийте найпростіший текстовий редактор (Блокнот, Windows Notepad, Notepad++ тощо).
2. Створіть у текстовому редакторі файл довільного змісту. Текст повинен мати заголовок, складатися з кількох абзаців, мати коментарі.
3. Збережіть розроблений код під назвою file1.html у папці на робочому столі комп'ютера.
4. Відкрийте файл file1.html за допомогою браузера Google Chrome або іншого браузера, який ви використовуєте.
5. Для перегляду розробленої веб-сторінки у майбутньому потрібно відкрити свою робочу папку і двічі натиснути мишею на назві файлу. Перегляд у браузері відбуватиметься за умовчанням.
6. Виконайте форматування тексту на створеній веб-сторінці. Для цього у файлі file1.html застосуйте різні накреслення літер тексту (жирний, курсив, підкреслений). Заголовок тексту розмістіть по центру та відокремте від іншого тексту порожнім рядком. Кожний абзац тексту розташуйте з нового рядка.
7. Додайте до тексту в файлі file1.html нумерований або маркований список. Спробуйте виокремити список у тексті іншим шрифтом.
8. Збережіть зміни, внесені до файлу і перегляньте оновлену веб-сторінку у браузері.
9. Зробіть висновок по роботі.

Домашнє завдання:

Створіть ще один html-файл і збережіть його під назвою file2.html. В тексті на веб-сторінці використовуйте якнайбільше тегів форматування і надайте своїй сторінці бажаного вигляду. Перегляньте результати своєї роботи у браузері.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Наведіть загальний вигляд структури html-документа.
2. Розкажіть, яке призначення тега та які параметри він може приймати.

3. Визначте, у яких тегах можна вставити заголовки на веб-сторінці.
4. Представте html-код, що задає виведення тексту у два рядки.
5. Опишіть загальний вигляд коду для порожньої веб-сторінки.
6. Перелічіть атрибути, що визначають тип вирівнювання.
7. Розкажіть, які види списків існують в html і якими тегамі вони задаються.

Література: [1] с. 39–41; [2] с. 45–51.

ЛАБОРАТОРНА РОБОТА № 4.

Тема: Задання кольорів та розміщення зображень на веб-сторінці.

Мета роботи: Навчитися задавати кольори для різних елементів веб-сторінки та вставляти зображення.

Короткі теоретичні відомості

Кольори на веб-сторінці можна задавати різними способами: використовуючи шістнадцяткове значення кольору у системі RGB або використовуючи константу кольору. Колір тексту на веб-сторінці можна задавати за допомогою відповідного значення атрибуту color, який вказується у відкриваючому тегі . Інший спосіб встановлення кольору тексту для всієї веб-сторінки полягає у застосуванні атрибуту text всередині тегу <body>, у якому міститься основне наповнення веб-сторінки.

Колір фону веб-сторінки можна задавати за допомогою атрибуту bgcolor тегу <body>, а використовуючи інший атрибут цього тегу background, можна встановити фонове зображення.

Зображення вставляються на веб-сторінку за допомогою одиночного тегу . Всередині тегу обов'язково повинен бути розміщений атрибут src із зазначенням місцезнаходження файлу із зображенням, що розміщується на веб-сторінці. Важливо, щоб шлях до графічного файлу із зображенням було прописано правильно із врахуванням усіх вкладених папок, всередині яких міститься файл. У випадку розміщення файлу із зовнішнього веб-ресурсу, необхідно зазначити його повну зовнішню адресу.

Важливо також, щоб формат графічного файлу дозволяв представляти його як зображення на веб-сторінці. Можна розміщувати зображення у форматі JPG, GIF, PNG для растрової графіки та SVG для векторної.

Ширина та висота зображення задається атрибутами width і height. Вирівнювання зображення на веб-сторінці здійснюється шляхом додавання відповідного значення до атрибуту align. Атрибути hspace та vspace дозволяють задати відстань від зображення до тексту по горизонталі та вертикалі відповідно. Рамка навколо зображення створюється за допомогою атрибуту border.

Атрибути alt і title тегу задають текст, який буде виводитися на місці, де має з'явитись зображення, до його завантаження або при відключеній графіці, а також якщо зображення завантажити не вдалося. Крім того, при наведенні курсору миші на зображенні, текст, вказаний в параметрі title, можна побачити у якості тексту спливаючої підказки.

Завдання: Задати колір тексту на веб-сторінці. Виконати заливку фону веб-сторінки кольором. Вставити зображення на веб-сторінку із текстом, задавши правильні параметри вирівнювання. Зробити зображення фоном веб-сторінки.

Хід роботи

1. Відкрийте файл із текстом, створений у попередній лабораторній роботі.
2. Змінюючи атрибути text і bgcolor тегу <body>, задайте колір фону і тексту на веб-сторінці.
3. Для окремої ділянки тексту задайте інший колір, відмінний від основного, за допомогою атрибуту color у тезі .
4. Спробуйте задавати кольори як за допомогою константи кольору, так і використовуючи код кольору у системі RGB.
5. Оберіть графічний файл на своєму комп'ютері у форматі JPG, GIF або PNG. Розмістіть його як зображення на веб-сторінці.
6. Встановіть розміри зображення та задайте параметри його вирівнювання відповідно до тексту на сторінці.
7. Візьміть зображення у рамку, встановіть її колір і ширину.
8. Встановіть текст спливаючої підказки при наведенні на зображення.
9. Для іншої веб-сторінки, розробленої у попередній роботі, встановіть фонове зображення за допомогою атрибуту background тегу <body> .
10. Зробіть висновок по роботі.

Домашнє завдання:

Створіть веб-сторінку, що містить декілька зображень із різними параметрами вирівнювання відповідно до тексту на сторінці. Слідкуйте за правильним записом шляху до графічного файлу в атрибуті src. Перегляньте результати своєї роботи у браузері.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Визначте тег, яким задається колір тексту на веб-сторінці.
2. Поясніть, як вставити зображення на веб-сторінку.
3. Назвіть атрибути, що задають розміри зображення на веб-сторінці.

4. Згадайте, які графічні формати дозволяють розміщувати зображення на веб-сторінці.
5. Перелічіть можливі значення атрибуту align для вирівнювання зображення відповідно до тексту на веб-сторінці.

Література: [1] с. 42–47; [2] с. 121–128.

ЛАБОРАТОРНА РОБОТА № 5.

Тема: Створення веб-сторінки з гіперпосиланнями та динамічними ефектами.

Мета роботи: Навчитися вставляти гіперпосилання на веб-сторінку, об'єднувати сторінки через перелінковку та створювати динамічні ефекти.

Короткі теоретичні відомості

Гіперпосилання – це вказівка для веб-браузера, до якого об'єкту в межах чи поза межами html-документа йому слід звернутися. За допомогою гіперпосилань користувачі можуть переходити з однієї веб-сторінки на іншу, завантажувати файли тощо. В якості гіперпосилання може бути оформлений фрагмент тексту або зображення. Коли веб-сторінка відображається у вікні веб-браузера, текстове посилання зазвичай виділене синім кольором і підкреслюванням. Для переходу на сторінку, до якої веде посилання, користувачу достатньо натиснути кнопкою миші на текст посилання.

Для створення гіперпосилання необхідно використовувати теги `<a>` та `</a>` в кодї веб-сторінки, визначивши для відкриваючого тегу `<a>` атрибут `href`. Його значенням має бути вказаний шлях до html-файлу веб-сторінки, на яку веде посилання. Текст посилання розміщують між тегами `<a>` та `</a>`.

Так само, як і у випадку зображень, важливо правильно прописувати місцезнаходження файлу html, на який відбувається перехід, із врахуванням усіх вкладених папок, всередині яких міститься файл. Якщо веб-сторінка, на яку вказує посилання, розміщена на іншому веб-сайті, то значенням атрибуту `href` повинна бути повна URL-адреса з назвою протоколу включно.

У якості посилання може слугувати не лише текст, але й зображення. Для створення посилання через зображення всередині тегів `<a>...</a>` слід записати тег `<img>`, в атрибуті `src` якого зазначити шлях до файлу із зображенням.

Для створення найпростішого динамічного ефекту на веб-сторінці можна використовувати біжучий рядок. Розміщувати біжучий рядок можна в будь-якому місці веб-сторінки за допомогою парного тегу `<marquee>`, який змушує текст всередині рухатися в тому чи іншому напрямку.

Завдання: Вставити текстові гіперпосилання на веб-сторінку. Створити посилання через зображення. Об'єднати розроблені сторінки, сформувавши навігаційну панель як список посилань. Розмістити біжучий рядок на веб-сторінці.

Хід роботи

1. Відкрийте один з html-файлів, розроблених вами у попередніх лабораторних роботах або створіть новий файл під назвою file5.html.
2. Вставте у свій файл текстове гіперпосилання на іншу веб-сторінку. Зверніть увагу на правильність вказаного шляху до html-файлу, на який здійснюється перехід.
3. Створіть гіперпосилання у вигляді графічного зображення. При натисканні на таке зображення з однієї веб-сторінки повинен здійснюватись перехід на іншу сторінку, збережену раніше під назвою file1.html.
4. Наприкінці веб-сторінки розташуйте рядок з гіперпосиланням для повернення на початок сторінки.
5. В лівій частині веб-сторінки сформууйте навігаційну панель у вигляді списку посилань.
6. До заголовку на веб-сторінці застосуйте динамічний ефект, задавши налаштування напрямку та швидкості прокручування.
7. Зробіть висновок по роботі.

Домашнє завдання:

Створіть веб-сторінку, що містить навігаційну панель у вигляді списку посилань, об'єднавши таким чином усі розроблені вами веб-сторінки. Наприкінці сторінки додайте біжучий рядок із текстом “Ця сторінка створена (*вказіть ваше Прізвище І.П.*)” на фоні смуги довільного кольору. Розділіть отриману веб-сторінку за допомогою горизонтальних ліній. Перегляньте результати своєї роботи у браузері.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Опишіть, як вставити гіперпосилання на веб-сторінку.
2. Покажіть, яким чином графічне зображення зробити гіперпосиланням.
3. Вкажіть, за допомогою якого тегу задається мітка на веб-сторінці.
4. Поясніть призначення параметрів link, vlink, alink.
5. Перелічіть можливі види посилань у межах однієї веб-сторінки.

6. Дайте визначення, що таке якір і яким чином задати якір на веб-сторінці.
7. Згадайте, які атрибути може мати тег `<marquee>`.

Література: [1] с. 65–68; [3] с. 137–147.

ЛАБОРАТОРНА РОБОТА № 6.

Тема: Розробка веб-сторінки на основі табличної структури.

Мета роботи: Вміти додавати таблиці на веб-сторінку та створювати на її основі табличний дизайн сайту.

Короткі теоретичні відомості

Таблиця на веб-сторінці створюється за допомогою парного тегу `<table>`, що задає початок і кінець таблиці. Для виділення рядків та стовпців, з яких складається таблиця, існують теги `<tr>` і `<td>`. Кожен рядок таблиці оголошується тегом `<tr>`, який додатково може містити параметри `align` для горизонтального вирівнювання тексту в комірках таблиці та `valign` для вертикального вирівнювання. Стовпчики таблиці позначаються тегом `<td>` всередині кожного із рядків.

Окрема комірка таблиці може мати свої власні значення параметрів `align` та `valign`, `width` та `height`, а також `bgcolor`. Крім того, всередині тегу `<tr>` можна вказувати атрибути `colspan` та `rowspan`. За допомогою цих атрибутів дві чи декілька комірок таблиці об'єднуються по горизонталі або вертикалі відповідно. Тег `<th>` дозволяє вказати комірки із назвами стовпців таблиці. Вміст таких комірок виділяється напівжирним шрифтом і розміщується по центру. В усьому іншому тег `<th>` аналогічний тегові `<td>`.

Заголовок усієї таблиці створюється за допомогою тегу `<caption>`. Атрибут `align` цього тегу може приймати одне з двох значень: `top`, коли заголовок записується над таблицею і `bottom`, коли заголовок розміщується під таблицею.

Рамка навколо таблиці створюється за допомогою атрибуту `border` тегу `<table>`. Товщину ліній сітки всередині таблиці можна регулювати, задаючи відповідну кількість пікселів як значення атрибуту `cellspacing`. Ширину самої таблиці можна вказувати в параметрі `width` у пікселях або в процентах від розміру вікна веб-сторінки.

В html-документах таблиці використовуються або у явному вигляді як засіб представлення даних, або як елемент оформлення сторінки, за допомогою якого можна в потрібному місці розмістити на веб-сторінці текст і графіку. Таблична структура веб-сторінки передбачає вставку html-таблиці на всю ширину сторінки. В комірках цієї таблиці розміщується контент. Таким чином, візуально веб-сторінку можна розділити на необхідну кількість частин: шапка сайту, панель навігації тощо. Табличний дизайн сайтів є

дуже зручним і поширеним, особливо в середовищі розробників-початківців. Основною перевагою такого підходу є його зрозумілість та можливість уникнути проблем сумісності.

Завдання: Вставити таблицю на веб-сторінку. Встановити ширину таблиці та її комірок. Оформити рамку навколо таблиці. Провести об'єднання комірок по горизонталі і по вертикалі. Розробити веб-сторінку на основі табличної структури, розмістивши елементи сторінки у комірках таблиці.

Хід роботи

1. Створіть нову веб-сторінку у файлі file61.html і розмістіть на ній таблицю з даними про свій поточний рейтинг. Таблиця буде складатися з 3 стовпців та 4 рядків.
2. Задайте заголовок таблиці та її ширину.
3. Вкажіть назви стовпців та рядків таблиці.
4. Змініть вигляд таблиці. Встановіть товщину її рамки, задайте кольори рамки та фону таблиці. Приклад таблиці, яка має з'явитися в результаті, наведений на рисунку:

Мої рейтинги

	Оцінка за семестр	Оцінка за рік
Архітектурна графіка	50	100
Архітектурний дизайн	50	100
Фінансова грамотність	50	?

5. Вирівняйте текст у клітинках таблиці. У комірках з назвами стовпців та рядків розміщення виконайте по центру, в інших клітинках – на ваш вибір.
6. Об'єднайте довільні дві клітинки таблиці в одну. В результаті вміст таблиці зміститься, як показано в наступному прикладі:

Мої рейтинги

	Оцінка за семестр	Оцінка за рік
Архітектурна графіка	50	100
Архітектурний дизайн	50	100
Фінансова грамотність	50	?

7. Приберіть зайві клітинки таблиці. Перегляньте результати своєї роботи у браузері.
8. Відкрийте нову веб-сторінку у файлі file62.html та створіть таблицю для розміщення тексту та зображення. Таблиця повинна складатися з одного рядка та двох стовпців: у першій комірці потрібно вставити зображення, у другій – текст. Після редагування властивостей, таблиця матиме загальний вигляд, як показано у наступному прикладі:



9. Зробіть висновок по роботі.

Домашнє завдання:

Створіть веб-сторінку file63.html на основі табличного підходу до дизайну сайту.

Об'єднайте розроблену сторінку з іншими сторінками за допомогою гіперпосилань. Зробіть зображення на сторінці file62.html гіперпосиланням на створений файл. Перегляньте результати своєї роботи у браузері.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Розкрийте призначення тегу <table> та які атрибути він може приймати.
2. Назвіть теги, які формують у таблиці рядки, клітинки-заголовки і звичайні клітинки.
3. Згадайте, за допомогою якого тегу створюється заголовок таблиці.
4. Вкажіть, за допомогою яких атрибутів створюється рамка навколо таблиці.
5. Опишіть структуру html-документу при табличному дизайні веб-сторінок.
6. Поясніть, яким чином об'єднати декілька комірок таблиці в одну.
7. Вкажіть атрибут, який використовують для вирівнювання елементів в комірках таблиці.
8. Подумайте, як можна регулювати товщину ліній сітки всередині таблиці.

Література: [1] с. 18–28, 48–51; [2] с. 61–74.

ЛАБОРАТОРНА РОБОТА № 7.

Тема: Створення сайтів на основі фреймової структури.

Мета роботи: Вміти створювати сайти, розподіляти їх на фрейми, створювати графічні навігаційні панелі.

Короткі теоретичні відомості

Зазвичай для всіх сторінок сайту деякі їхні частини, такі як назва сайту зверху і бічна панель навігації, містять одну й ту саму інформацію. При переході між сторінками змінюється лише основне наповнення кожної сторінки. Відтак, залишаючи в незмінному стані вміст однакових частин, при переходах між сторінками сайту можна завантажувати лише наповнення основної області. Інструментом, який дає змогу реалізувати таке рішення, є створення фреймової структури.

Фрейми дозволяють розбити вікно веб-браузера на декілька прямокутних областей, в кожен з яких можна завантажити окремий html-файл. Отже, фрейми створені для того, щоб визначити робочі області для кожного html-документа в єдиному вікні веб-браузера.

Для опису фреймової структури потрібно використовувати парний тег `<frameset>`. Всередині тегів `<frameset>` та `</frameset>` можуть міститись теги `<frame>`, `<iframe>` або інший набір фреймів, описаний тегами `<frameset>` та `</frameset>`.

Тег `<frameset>` може мати атрибути `rows` та `cols`, що вказують браузеру, яким чином слід розміщувати фрейми у його вікні. Атрибут `rows` описує розбиття на горизонтальні фреймові області, атрибут `cols` – розбиття на вертикальні області. В якості значень параметрів `rows` та `cols` вказуються розміри фреймів. Розміри можуть бути зазначені в абсолютних одиницях (у пікселях) або у відсотках.

Тег `<frame>` або `<iframe>` описує окремий фрейм і містить обов'язковий атрибут `src`, що визначає шлях до html-файлу, який повинен бути завантажений у фреймову область. Використання фреймової структури передбачає одночасне відображення вмісту різних веб-сторінок, адреси яких прописуються у параметрах фрейму.

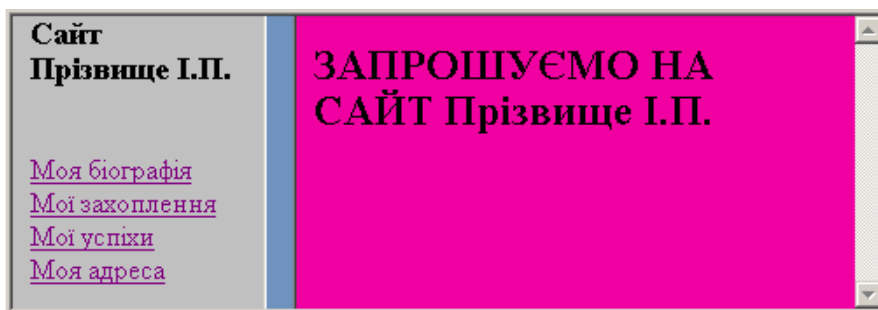
Карта зображень – це графічна панель на веб-сторінці з активними областями, через які можна здійснювати переходи на інші сторінки. Карти зображень створюються як

навігаційні карти сайту. При цьому зображенню на веб-сторінці за допомогою атрибуту usemap тегу присвоюється індивідуальне ім'я. Навігація через зображення визначається тегом <map>, у якого в атрибуті name слід вказати назву зображення. Активні області на карті зображень описуються тегом <area>, у параметрах якого задається форма кожної області, її координати, адреса файлу, на який здійснюється перехід через цю область. Карти зображень дозволяють реалізувати творчий дизайнерський підхід до створення навігаційної панелі сайту.

Завдання: Розробити фреймову структуру сайту на основі простої і складної конфігурації фреймів. Створити карту зображень. Розмістити карту зображень у якості графічної навігаційної панелі розробленого сайту.

Хід роботи

1. Створіть нові або оберіть серед html-файлів, розроблених вами у попередніх лабораторних роботах, файл із основним текстом веб-сторінки та файл з навігаційною панеллю.
2. Напишіть код html, який буде містити структуру сайту та розподілятиме його на фрейми.
3. Вікно веб-браузера розділіть на два вертикальних фрейми: більш вузький з лівого боку і більш широкий праворуч. Лівий фрейм міститиме перелік посилань для навігації по сайту, правий слугуватиме як початкова сторінка сайту.
4. Дєднайте обрані файли зі змістом у ліву та праву фреймові області. В результаті має з'явитися структура веб-сторінки наступного вигляду:

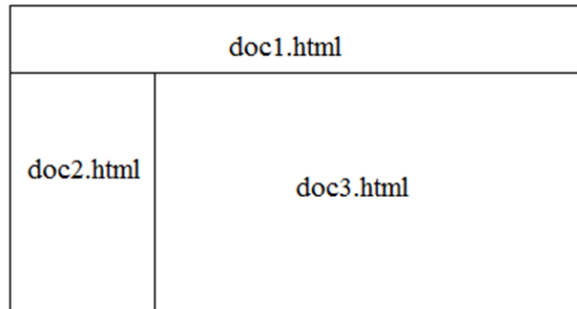


5. Відредагуйте параметри відображення фреймів: додайте смугу прокручування для правого фрейму, задайте різний колір фону для областей, приберіть межу між фреймами.
6. Перегляньте отриманий результат у браузері.
7. Оберіть зображення для графічної навігаційної панелі.
8. Створіть html-файл для карти зображень. Задайте відповідні розміри карти зображень. Опишіть активні області для переходу на інші сторінки.
9. Збережіть файл під назвою karta.html.. Відкрийте його у браузері.
10. Змініть файл із описом фреймової структури таким чином, щоб у лівому фреймі було розміщено створену карту зображень.

11. Зробіть висновок по роботі.

Домашнє завдання:

Створіть html-документ, що описує фреймову структуру за наступною схемою:



Для цього вікно веб-браузера поділіть на два горизонтальних фрейми. Нижню частину розділіть по вертикалі на лівий та правий фрейми, які займають, відповідно 20% та 80% загальної ширини веб-сторінки. Активні області розробленої карти зображень зробіть у вигляді різних геометричних фігур (наприклад, кола, прямокутника та трикутника). Об'єднайте файли у новій конфігурації фреймової структури.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Назвіть тег, який здійснює розподіл фреймової структури.
2. Вкажіть, який тег та які атрибути описують окремі фрейми.
3. Поясніть, як створюються складні конфігурації фреймів.
4. Розкрийте, як представити зображення у вигляді навігаційної карти.
5. Опишіть, яку будову має тег-контейнер <map>.
6. Визначте, що таке гаряча область на карті зображень і яким чином задати різні форми таких областей.
7. Згадайте, яке призначення має параметр name тегу <map>.
8. Поясніть, що таке холодний фрагмент на карті зображень.
9. Розкрийте, яке призначення має тег-контейнер <noframes>...</noframes>.

Література: [1] с. 51–59; [2] с. 74–80.

ЛАБОРАТОРНА РОБОТА № 8.

Тема: Додавання форми на веб-сторінку.

Мета роботи: Навчитися створювати сторінки з кнопками, списками, полями введення та іншими елементами форм.

Короткі теоретичні відомості

Форма на веб-сторінці представляє собою документ, створений за допомогою елементів HTML. Призначенням форми є збір інформації від користувачів. Для створення форми в html-документі застосовується парний тег `<form>...</form>`. Даний тег встановлює межі використання інших елементів, розміщених всередині форми.

Для визначення області всередині форми, де здійснюється збір даних, використовується елемент `<input>`. Цей елемент представляє собою поле для введення інформації користувачем. В атрибуті `type` тегу `<input>` слід вказати вид елементу форми, який потрібно вставити на веб-сторінку. Значення `text` дозволяє створювати однорядкове текстове поле. Значення `password` відповідає за поле для введення паролю. Для вставки елементів-прапорців у формах можна використовувати значення `checkbox` атрибуту `type`. Такий елемент передбачає можливість відмітити декілька пунктів в переліку значень. Вибір лише одного варіанту з кількох можливих здійснюється через елементи-перемикачі, які задаються за допомогою компоненти `radio`. Створення кнопки, яка повертає початкові значення полів усіх елементів форми, відбувається шляхом додавання елементу `reset`. Кнопка `submit` використовується при завершенні введення користувачем інформації і відправці даних на сервер.

Багаторядкове текстове поле, яке передбачає введення великої кількості друкованої інформації у декілька рядків в межах форми, створюється за допомогою парного тегу `<textarea>`. Для організації списків з прокручуванням та випадających меню можна використовувати тег `<select>`. Визначення пунктів такого списку відбувається через теги `<option>` всередині конструкції `<select>... </select>`.

Після того, як користувач заповнить форму і запускає процес її обробки, інформація з неї потрапляє в програму, що працює на сервері.

Завдання: Розмістити форму в html-документі. Створити різні елементи управління — кнопки, списки, елементи-перемикачі, прапорці, поля введення тощо. Вказати властивості різних елементів форми.

Хід роботи

1. Створіть сторінку anketa.html. Оголосіть розміщення форми на веб-сторінці.
2. Вставте однорядкове текстове поле з обмеженням до 10 символів та поле для введення паролю, який буде складатися з 5 символів.
3. Розташуйте список елементів-перемикачів, де перший пункт відмічено за умовчанням.
4. Для вибору кількох можливих значень з переліку, створіть набір елементів checkbox, згрупованих шляхом встановлення однакових значень атрибуту name.
5. Створіть випадаючий список з 5 пунктів, 3 з яких є видимими. Останній пункт списку оберіть за умовчанням. Передбачте багатоваріантний вибір.
6. Розташуйте багаторядкове текстове поле з 5 рядків та 40 символів, зі впливаючою підказкою.
7. Наприкінці форми додайте кнопки reset і submit. Встановіть підписи на кнопках українською мовою.
8. Після закриття форми, додайте гіперпосилання на довільну веб-сторінку.
9. Перегляньте отриманий результат у браузері. Приклад форми, яка має з'явитися в результаті, наведений на рисунку:

Анкета

Заповни анкету для вступу в клуб "Віртуальна бібліотека"

Введіть повне ім'я:

Введіть пароль:

Стать:

- ☐ Чоловіча
☐ Жіноча

Вид діяльності:

- ☐ Студент
☐ Школяр
☐ Аспірант
☐ Працюю ☐ Не працюю

Які книги ти любила/любив читати?

Науково-познавальна
Фантастика
Фентезі

Що читаєш зараз?

Це посилання на сторінку "Бібліотека":

[Mail google](#)

10. Зробіть висновок по роботі.

Домашнє завдання:

Відформатуйте створену веб-сторінку з формою для збору інформації. Задайте колір фону і тексту, шрифт та спосіб накреслення для різних елементів. Виділіть заголовки та вкажіть спосіб вирівнювання. Перегляньте результати своєї роботи у браузері.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Розкажіть, для чого використовуються форми.
2. Згадайте, за допомогою яких тегів створюються форми.
3. Назвіть спосіб застосування різних елементів форми.
4. Поясніть, як створити текстове поле для введення інформації.
5. Розкрийте, як у випадючих списках встановити значення за умовчанням.
6. Опишіть спосіб зберігання даних, введених у форму.

7. Вкажіть, як розмістити прапорці та перемикачі у веб-формі.
8. Дайте визначення, що собою являють елементи checkbox та radio.
9. Поясніть, яким чином об'єднати декілька елементів прапорців або перемикачів в одну групу.
10. Розкрийте призначення кнопок reset і submit.
11. Перелічіть відомі вам атрибути тегу <textarea>.

Література: [1] с. 59–65; [5] с. 194–207.

ЛАБОРАТОРНА РОБОТА № 9.

Тема: Знайомство з таблицями стилів.

Мета роботи: Оволодіти навичками оформлення веб-сторінок, використовуючи каскадні таблиці стилів.

Короткі теоретичні відомості

Каскадні таблиці стилів CSS (Cascading Style Sheets) – це набір правил оформлення та форматування, який може бути застосовано до різних елементів веб-сторінки. Використовуючи CSS, можна описати властивості елементів та визначити цей опис як

стиль, і в подальшому вказувати, що елемент, який ви бажаєте оформити певним чином, повинен прийняти властивості вже описаного стилю.

Властивість стилю – це параметр стильового оформлення документу, який визначається специфікацією CSS. Елемент, якому належить визначений стиль, узагальнено називається селектором. Селектори поділяються на декілька типів: селектори тегів, ідентифікатори та класи. При заданні стилю після запису селектора йдуть фігурні дужки, в яких зазначається потрібна стильова властивість, а її значення вказується після двокрапки. У якості селектора може виступати будь-який тег HTML, для якого визначаються правила форматування, такі як колір, фон, розмір тощо.

Таким чином, в наборі правил CSS спочатку вказується ім'я тегу, оформлення якого буде перевизначено, без дужок, великими чи малими буквами. Потім всередині фігурних дужок записується властивість CSS, а після двокрапки – її значення. Набір параметрів розділяється між собою крапкою з комою і може розміщуватись як в один рядок, так і в декілька. Таблиці стилів – це, фактично, набори CSS-правил, які задають властивості форматування елементів веб-сторінки.

Стилі CSS можуть включатися в html-документ 3 різними способами: внутрішні стилі, стилі рівня документу і зовнішні стилі. Внутрішній стиль записується в атрибуті style та застосовується лише до вмісту зазначеного тегу. Внутрішній стиль має більш високий пріоритет, аніж зовнішні стилі та стилі рівня документу, проте такий спосіб задання стилю не відповідає принципу розділення змісту веб-сторінки та її оформлення.

Стилі рівня документу застосовуються до всіх однакових елементів на веб-сторінці, записуються всередині тегу `<style>...</style>`, який вкладається в тег `<head>...</head>` у структурі коду HTML. Такий спосіб задання стилів використовується, коли потрібно застосувати однакові стилі одразу до багатьох тегів в одному документі.

У зовнішніх файлах потрібно зберігати стилі, спільні для всього сайту, вони впливають одразу на велику кількість тегів у багатьох документах. Це виявляється дуже зручним, якщо сайт містить багато сторінок. При застосуванні зовнішньої таблиці стилів опис селекторів та їхніх властивостей розміщується в окремому файлі з розширенням .css, а для того, щоб зв'язати html-документ з цим файлом, використовується тег `<link>`. Даний тег поміщається в контейнер `<head>...</head>`. Значення атрибутів rel та type тегу `<link>` залишаються незмінними, атрибут href задає шлях до CSS-файлу зі стилями. Використання зовнішнього файлу зі стилями дозволяє спростити задачу редагування стилів для великих сайтів. Якщо усі сторінки підключають один і той самий зовнішній стиль CSS, достатньо в ньому задати нове значення властивості для певного виду елементів на всіх сторінках. Інакше довелося б редагувати кожен елемент окремо.

Завдання: Застосувати визначення стилю в тегах. Розмістити опис стилів у заголовку html-документу за допомогою тегів `<style>...</style>`. Підключити зовнішню таблицю стилів. Здійснити форматування шрифту та тексту за допомогою мови CSS.

Хід роботи

1. Створіть сторінку file9.html. Розмістіть на веб-сторінці довільний текст, заголовки, гіперпосилання і зображення.
2. Застосуйте внутрішній стиль до одного з параграфів тексту на сторінці: за допомогою атрибуту style задайте шрифт в цьому параграфі Georgia, колір тексту фіолетовий, накреслення курсивом.

3. Заголовки на сторінці відобразить шрифтом Arial, розміром 14 pt, червоного кольору. Зробити це можна, задавши стилі рівня документу всередині тегу `<style>...</style>`.
4. Додajte до веб-сторінки зовнішню таблицю стилів під назвою `mystyle.css`.
5. У зовнішній таблиці стилів задайте жовтий колір фону веб-сторінки.
6. Шрифти встановіть Arial або Times New Roman, колір тексту блакитний, відступи до тексту зверху та знизу по 20 пт, ліве та праве поля – по 10 пт.
7. Змініть колір гіперпосилань, заданий за умовчанням. Посилання до переходу по ним нехай відображатимуться червоним кольором, після переходу – зеленим.
8. Перегляньте отриманий результат у браузері.
9. Зробіть висновок по роботі.

Домашнє завдання:

Подумайте, яким чином для різних посилань на веб-сторінці використовувати різні кольори. Спробуйте реалізувати це у коді, прибравши підкреслення у посилань, задане за умовчанням.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді `html`-файлів та створених файлів зі стилями `css`. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Опишіть особливості застосування таблиць каскадних стилів.
2. Розкрийте сутність основної концепції `CSS`.
3. Вкажіть способи зв'язку каскадних стилів з `html`-документом.
4. Перелічіть одиниці вимірювання у `CSS`.
5. Вкажіть псевдо-клас, що дозволяє оформити посилання, на які наведений курсор миші.

Література: [1] с. 69–74; [5] с. 118–144, 157–183.

ЛАБОРАТОРНА РОБОТА № 10.

Тема: Управління відображенням елементів веб-сторінки.

Мета роботи: Вміти створювати веб-сторінки з різним відображенням контенту для досягнення конкретних цілей веб-дизайну.

Короткі теоретичні відомості

Встановлення властивостей елементів, розміщених на веб-сторінці, засобами CSS передбачає застосування стилів до обмежених областей, які називають блоками-контейнерами, та їхнє позиціонування.

Блок-контейнер – це невидима прямокутна область, що визначена браузером. Таблиці стилів дозволяють управляти цією областю, встановлюючи її положення на сторінці з використанням абсолютних або відносних значень позиціонування.

Позиціонування блоку за допомогою властивості position за замовчуванням має значення static, іншими значеннями є absolute (абсолютне), relative (відносне) і fixed (фіксоване) позиціонування.

Абсолютне позиціонування дозволяє задавати координати внутрішнього блоку щодо зовнішнього контейнера, яким може бути батьківський елемент документа або інший блок відповідно до вікна перегляду браузера. При цьому блок елемента повністю видаляється з потоку документа й позиціонується щодо його блока-контейнера.

Відносне позиціонування дозволяє задавати координати блоку щодо його незміщеного положення в потоці документа. При відносному позиціонуванні елемент зміщується зі свого звичайного положення, але простір, який він повинен був займати, не зникає.

Фіксоване позиціонування дозволяє задавати координати блоку щодо початкового вікна перегляду браузера. Фіксований елемент прив'язується до певної точки у вікні браузера й не переміщується, навіть при прокручуванні веб-сторінки. Такі елементи часто застосовують для створення навігаційних панелей та рекламних блоків.

Порядок розташування блоків на веб-сторінці визначає z-index. Значеннями z-індексу є цілі числа (позитивні й негативні), причому блоки з більшими значеннями z-індексу будуть з'являтися над блоками з меншими значеннями.

Елемент <div> використовується для визначення властивостей прошарку. Він є контейнерним елементом і дозволяє застосувати позиціонування й z-index для одного або кількох елементів, розміщених у ньому.

До появи останніх стандартів CSS для обтікання текстом зображень користувалися атрибутом align елемента img зі значеннями left і right. Зараз стандартизована властивість float зі значеннями left і right може бути застосована не тільки до зображень, але й до інших елементів HTML. У комбінації з властивістю float часто застосовують властивість clear зі значеннями left, right або both для того, щоб звільнити місце ліворуч або праворуч від інших елементів веб-сторінки. Властивість float також можна використовувати при необхідності виведення тексту у декілька стовпців.

Завдання: Створити блоки на веб-сторінці. Наповнити їх контентом, встановити стиль рамок і зображень у блоках. Задати фон елементів через колір або фоновий малюнок. Застосувати властивість float для обтікання текстом зображення. Визначити абсолютне, відносне та фіксоване позиціонування для різних блоків. Встановити порядок накладення шарів елементів за допомогою властивості z-index.

Хід роботи

1. Створіть html-документ. Розмістіть на веб-сторінці зображення і довільний текст.
2. За допомогою CSS забезпечте використання плаваючої моделі для зображення.
3. Покажіть використання властивості float для створення документа, що містить 3 колонки.
4. Розмістіть декілька блоків на веб-сторінці. Наповніть їх контентом.
5. Продемонструйте приклад абсолютного позиціонування блоків.
6. Реалізуйте власний приклад блоку з фіксованим позиціонуванням.
7. Покажіть приклад відносного позиціонування блоку відносно батьківського елемента.
8. Застосовуючи властивість z-index, встановіть порядок накладення шарів для декількох блоків, що накладаються один на одного.
9. Зробіть висновок по роботі.

Домашнє завдання:

Задайте фон блоків-контейнерів через колір або фонове зображення. Встановіть різний стиль рамок для різних блоків. За допомогою властивостей margin і padding визначте відстань між блоками, а також поля всередині блоків.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів та створених файлів зі стилями css. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Охарактеризуйте CSS-властивості, за допомогою яких можна оформити границі елемента.
2. Розкрийте призначення тегів <div> та .
3. Опишіть, які види позиціонування елементів існують в CSS.
4. Назвіть CSS-властивість, за допомогою якої можна змінювати прозорість елементів.

Література: [1] с. 75–81; [5] с. 185–191.

ЛАБОРАТОРНА РОБОТА № 11.

Тема: Створення сайту на основі блочної структури.

Мета роботи: Вивчення принципів блокової верстки для вирішення задач побудови сайту.

Короткі теоретичні відомості

Блоковий підхід при верстці та дизайні сайтів полягає у використанні блокових елементів для розміщення контенту. Блоки-контейнери можуть відображати певний контент, і до кожного з них може бути застосований індивідуальний стиль. Основним елементом, що використовується в блоковій верстці, є тег `<div>`. Частина коду, відокремлена цим тегом, називається шаром. Тег `<div>` – це так званий контейнер (блок), який може містити форматований текст, зображення та ін. Верстка на основі `<div>` активно використовує CSS для гнучкого й зручного форматування елементів веб-сторінок. За допомогою CSS можна з точністю до пікселя задати розташування блоків на сторінці, необхідні відступи, колір, розмір шрифтів і багато іншого.

У новій версії мови гіпертекстової розмітки, HTML5, з'явилися нові теги для позначення блокових елементів для верстки сайту: `<header>`, `<footer>`, `<nav>`, `<aside>`, `<section>`, `<article>`, `<main>` тощо. Таким чином, різні частини веб-сайту отримали свої уніфіковані імена, що дозволяє уникнути плутанини між різними блоками.

Для того, щоб блокові елементи розташувалися разом по горизонталі, необхідно задати однакове значення властивості `float` для наступних один за одним елементів. З метою розміщення блоків нижче вирівняних по горизонталі, необхідно використовувати атрибут стилю `clear`, який вказує, що даний блок повинен розташовуватися нижче контейнерів, що передують йому.

Для керування відображенням вмісту контейнера у випадках, коли розмір контенту перевищує межі розмірів контейнера, використовується атрибут стилю `overflow`. Існує можливість управління відображенням вмісту контейнера окремо по горизонталі та вертикалі відповідно за допомогою атрибутів `overflow-x` і `overflow-y`, значення яких аналогічні значенням `overflow`.

При встановленні розмірів блоків слід правильно розраховувати їхню ширину з огляду на інші блоки, розташовані поруч. Загальна ширина екрану складає 100%, тобто щоб розмістити два блоки поруч, необхідно задати їхню ширину таким чином, щоб сума значень ширини двох блоків завжди дорівнювала 100%. Інакше може з'явитися пустий простір, або блоки не будуть вміщуватись на веб-сторінці, що порушить їхню структуру, тобто один з них може переміститися нижче або вище за інший, або зміститися поверх нього.

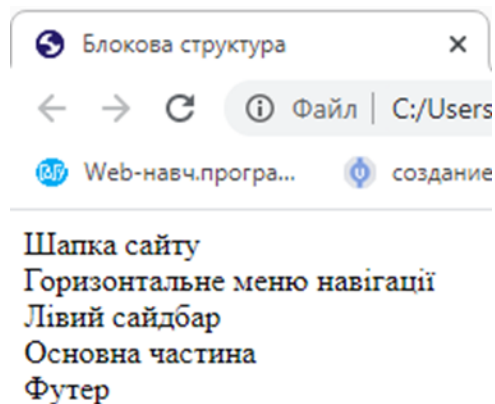
Завдання: Написати код веб-сторінки на основі блочного підходу до веб-дизайну. Створити таблицю каскадних стилів, яка відформатує розроблений html-документ.

Хід роботи

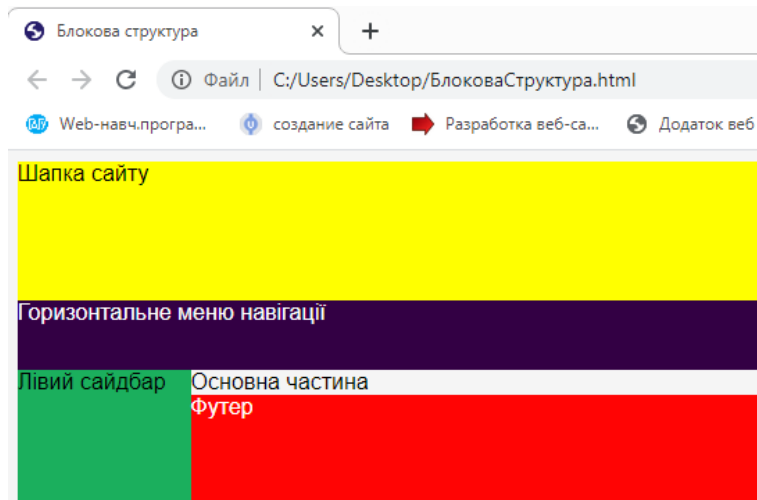
1. При розробці сайту з блоковою структурою передусім слід зрозуміти, який шаблон веб-сторінки нам потрібен, тобто намалювати її будову. У даній роботі будемо використовувати просту структуру, яка зустрічається у більшості сайтів. Вона складатиметься з шапки, горизонтального меню, лівого сайдбару, основної частини та футеру. Представимо необхідну будову сторінки графічно:



2. Напишіть html-код веб-сторінки, що задає блокову структуру сайту за допомогою тегів `<header>`, `<footer>`, `<nav>`, `<aside>`, `<main>`.
3. Без підключення стилів CSS представлений код у браузері виглядатиме наступним чином:



4. Для того, щоб веб-сторінка набула потрібного вигляду, додайте стилі `<style>...</style>` у розділі `<head>` створеного html-документу.
5. Перевірте результат у браузері. Після застосування стилів розроблюваний сайт набуває потрібного вигляду:



6. Наповніть блоки різними елементами: зображеннями, гіперпосиланнями, елементами форми, текстом тощо. Задайте вирівнювання елементів всередині блоку.
7. Зробіть висновок по роботі.

Домашнє завдання:

Розробіть власний макет сторінки на основі блоків-контейнерів `<div>`, використовуючи HTML та CSS. Стили приєднайте за допомогою зовнішнього файлу `style.css`. Представте декілька сторінок сайту за власним шаблоном блокової структури.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Відповідно до ходу роботи, необхідно представити коди розроблених веб-сторінок у вигляді html-файлів та створених файлів зі стилями `css`. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Розкажіть, що таке успадковування властивостей у CSS.
2. Перерахуйте призначення та застосування властивості прокручувань.
3. Опишіть структуру документу при блоковому дизайні веб-сторінок.
4. Вкажіть, за допомогою яких тегів реалізується блокова верстка сайтів.

Література: [2] с. 159–172; [5] с. 244–271.

ЛАБОРАТОРНА РОБОТА № 12.

Тема: Інтерфейс програми Figma. Базові інструменти дизайну.

Мета роботи: Набуття навичок роботи в програмному середовищі Figma.

Короткі теоретичні відомості

Figma – це сучасний хмарний інструмент для проєктування інтерфейсів, що використовується у веб-дизайні, UI/UX-дизайні та створенні прототипів. Програма працює за принципом візуального редактора, що дозволяє створювати макети веб-сайтів, мобільних додатків та інших цифрових продуктів у реальному часі. Figma є кросплатформовим рішенням, оскільки функціонує у веб-браузері та має додатки для операційних систем Windows, macOS та інших платформ.

Дане програмне забезпечення є універсальним інструментом для вирішення широкого спектра завдань у веб-дизайні та розробці інтерфейсів. Його функціонал дозволяє проєктувати інтерфейси різної складності: від веб-сайтів до мобільних додатків і десктопних програм.

Також одним із важливих аспектів роботи у Figma є забезпечення ефективної командної взаємодії. Програма дозволяє декільком учасникам команди одночасно працювати над одним проєктом, що значно спрощує процеси узгодження дизайну та комунікації між дизайнерами, розробниками та замовниками. Це особливо актуально для великих проєктів, де необхідно досягти єдиної візуальної та функціональної концепції. Функціонал Figma включає багато зручних та гнучких інструментів для проєктування інтерфейсів. Вони дозволяють швидко створювати фрейми, фігури, текстові блоки та інші елементи, що є необхідними для дизайну. Завдяки можливості створення інтерактивних прототипів, користувачі можуть тестувати роботу інтерфейсу та оцінювати його зручність на етапі проєктування. Це також дозволяє вчасно виявити недоліки і внести необхідні корективи ще до початку розробки.

Крім того, Figma підтримує створення дизайн-систем, що включають уніфіковані стилі, компоненти та шаблони. Це особливо важливо для великих проєктів, де необхідно підтримувати візуальну та функціональну цілісність продукту. Вбудовані можливості експорту дозволяють легко передавати графічні ресурси у різних форматах, а це спрощує співпрацю між дизайнерами та розробниками на фінальних етапах роботи.

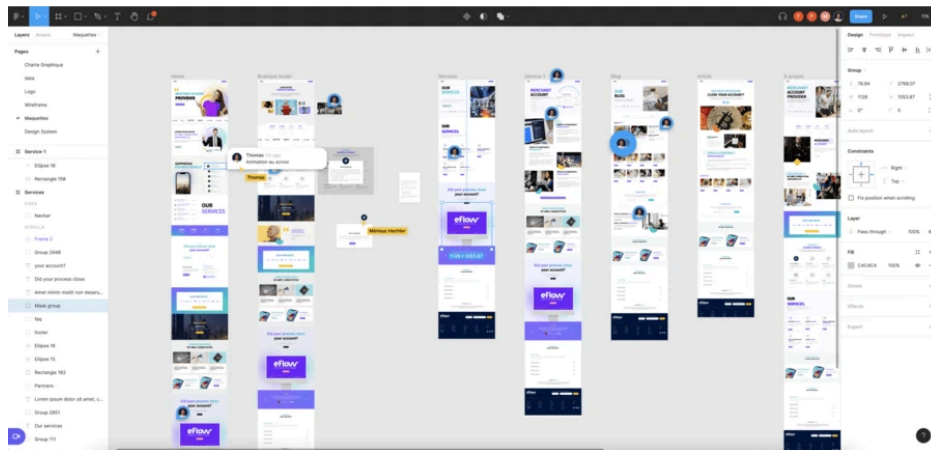


Рис. 12.1 – Інтерфейс програми Figma

При запуску програми Figma користувача зустрічає стартовий екран, який має чистий і структурований інтерфейс, орієнтований на швидкий доступ до основних функцій. У верхній частині екрана розташована панель навігації, де можна знайти поле для пошуку проєктів, а також доступ до профілю користувача та опцій створення нових файлів. У правому верхньому куті розташована кнопка "New", що відкриває меню для створення нового Design file, FigJam file, імпорту файлів або перегляду готових шаблонів.

Центральну частину стартового екрана займає блок "Recently viewed", де відображаються файли, з якими користувач працював нещодавно. Кожен файл представлений у вигляді картки з мініатюрою, що дозволяє швидко зорієнтуватися у вмісті. Під кожною карткою вказується назва файлу, статус (наприклад, "In Drafts") та час останнього редагування.

Ліва частина екрана містить бічну панель навігації, де зібрано основні категорії файлів і команд. Тут розташовані розділи "Recent" для швидкого доступу до нещодавно відкритих файлів, "Drafts", де зберігаються чернетки, а також "Community", що надає доступ до спільноти Figma для перегляду ресурсів та шаблонів. Також є можливість створити нову команду через опцію "Create new team".

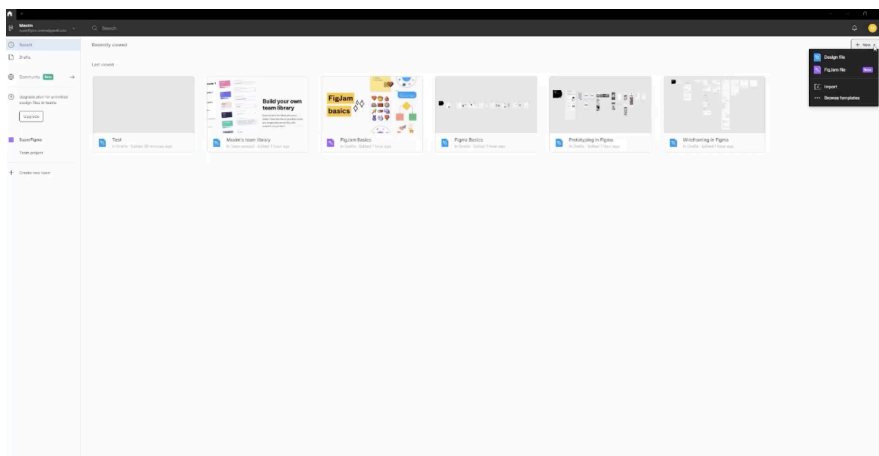


Рис. 12.2 – Стартовий екран програми Figma

При створенні нового проєкту відкривається екран програми з робочим інтерфейсом. У верхній частині екрана розташована головна панель інструментів, де зібрані базові функції для роботи. Тут можна знайти інструменти для створення фреймів, фігур, текстових блоків, а також опції для редагування та налаштування об'єктів. Крім цього, на панелі є кнопки для перемикання режимів перегляду та функція "Share" для спільного доступу до файлу.

Ліворуч розташована панель шарів і сторінок. У верхній частині цієї панелі відображається перелік сторінок проєкту, що дозволяє користувачеві керувати кількома робочими областями в межах одного файлу. У прикладі на рис. 12.3 створено три сторінки: Page 1, Page 2 і Page 3. Це дозволяє зручно організовувати проєкт і розділяти його на логічні частини. Під кожною сторінкою згодом відображатимуться шари (Layers), що містять усі елементи дизайну на обраній сторінці.

Центральна область є робочим полотном, де відбувається основна робота над дизайном. Полотно має необмежений простір, що дозволяє розміщувати фрейми, компоненти та інші елементи без обмежень. При першому запуску нового проєкту воно є порожнім.

З правого боку екрану розташована панель властивостей. Дана панель змінюється залежно від обраного інструмента чи елемента. У верхній частині цієї панелі можна бачити вкладки "Design", "Prototype" і "Inspect", які відповідають за налаштування дизайну, створення прототипів та перевірку властивостей для розробників. У нижній частині панелі можна налаштувати фон робочого полотна та знайти опції для експорту готових елементів.

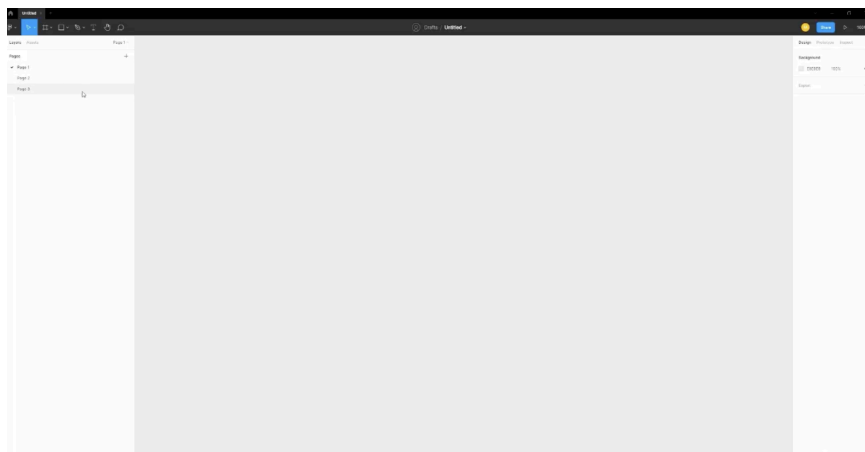


Рис. 12.3 – Робочий простір Figma

У верхній лівій частині вікна зібрані основні інструменти для роботи у Figma:

Move Tool – інструмент для переміщення об'єктів та зміни їх розмірів і позиції на робочому полотні. Використовується для маніпуляцій з будь-якими елементами дизайну, включно з фреймами, фігурами та текстом. Гаряча клавіша: V.

Frame Tool – використовується для створення фреймів, які є контейнерами для елементів дизайну, як-от сторінки, екрани або секції макету. Фрейми забезпечують структуроване розташування об'єктів та дозволяють налаштовувати розміри полотна. Гаряча клавіша: F.

Shape Tools – інструменти для побудови базових фігур: прямокутників, ліній, кіл та інших геометричних елементів. Вони є основою для створення графічних елементів дизайну або використовуються як допоміжні форми для складніших компонентів. Гарячі клавіші: R для прямокутника, L для лінії.

Pen Tool – інструмент для створення векторних кривих і контурів, що дозволяє малювати нестандартні фігури та форми з високою точністю. Широко застосовується для створення ілюстрацій, логотипів та складних елементів інтерфейсу. Гаряча клавіша: P.

Text Tool – інструмент для додавання текстових блоків у макеті. Використовується для створення заголовків, описів та іншого текстового контенту, необхідного для проектування інтерфейсу або прототипів. Текстові блоки можна налаштовувати за шрифтами, розмірами та кольорами. Гаряча клавіша: T.

Hand Tool – інструмент для навігації полотном шляхом його переміщення у потрібному напрямку. Полегшує роботу з великими макетами та дозволяє швидко переглядати різні ділянки робочої області. Гарячі клавіші: H або Пробіл (утримання).

Comment Tool – інструмент для додавання коментарів до об'єктів дизайну. Використовується для командної роботи, забезпечуючи можливість залишати зворотний зв'язок або уточнення безпосередньо у середовищі Figma. Гаряча клавіша: C.

Завдання: Створити новий проєкт у Figma. Розмістити фрейм та побудувати у ньому просту геометричну композицію з використанням базових інструментів програми.

Хід роботи

1. Відкриваємо програму Figma та створюємо новий проєкт.
2. За допомогою інструменту Frame Tool створюємо фрейм довільного розміру.
3. Використовуючи такі інструменти, як Shape Tools та Pen Tool (і різні параметри їх налаштування), а також інші допоміжні базові інструменти, створюємо графічну композицію. Приклад готової композиції зображено на рис. 12.4:

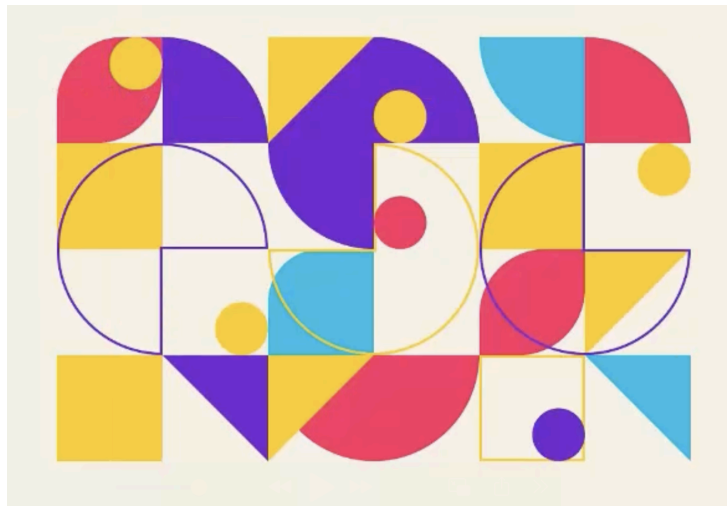


Рис. 12.4 – Графічна композиція, створена у Figma за допомогою базових інструментів

4. Зробіть висновок по роботі.

Домашнє завдання:

Ознайомтесь з інтерфейсом програми Figma та вивчіть базові інструменти для створення веб-дизайну.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Розкрийте, для чого використовується програма Figma і які її основні переваги.
2. Охарактеризуйте базові інструменти дизайну в програмному середовищі Figma.
3. Опишіть базові інструменти Figma для створення графічних об'єктів.

Література: [1] с. 172–185; [4] с. 11–15.

ЛАБОРАТОРНА РОБОТА № 13.

Тема: Робота з фреймами, групами, масками та обмеженнями у Figma.

Мета роботи: Навчитися оперувати інструментами, що відповідають за групування та стилізацію об'єктів.

Короткі теоретичні відомості

Фрейм – це базовий контейнер у Figma, що використовується для організації елементів дизайну на робочому полотні. Фрейми є невід'ємною частиною структури проєктів, оскільки вони дозволяють створювати окремі робочі області для компонентів, сторінок, секцій або повноцінних екранів інтерфейсу.

Фрейми виконують функцію контейнерів, які можуть містити інші об'єкти, такі як текст, фігури, зображення, групи та вкладені фрейми. Вони є аналогом “артбордів” у інших графічних редакторах, проте мають розширений функціонал, зокрема підтримку адаптивного дизайну та обмежень. У межах фрейму також можна задавати обмеження (Constraints), які регулюють розташування та поведінку об'єктів при зміні розмірів фрейму. Це особливо корисно для створення інтерфейсів, що адаптуються до різних розмірів екранів (мобільних пристроїв, планшетів, десктопів тощо).

Фрейми можуть містити інші фрейми або компоненти, що дозволяє створювати складні, багаторівневі структури. Фрейми також підтримують функцію експорту як окремих елементів дизайну у форматах PNG, SVG, JPG або PDF. Крім того, фрейми є основою для побудови інтерактивних прототипів, де кожен фрейм може виступати окремим екраном користувацького інтерфейсу.

Для створення фрейму необхідно обрати Frame Tool (гаряча клавіша: F) та визначити один із готових пресетів розмірів (наприклад, iPhone, Desktop, Tablet) або створити кастомний розмір. Фрейм з'явиться на полотні, і у нього можна додавати інші об'єкти. (Рис. 13.1)

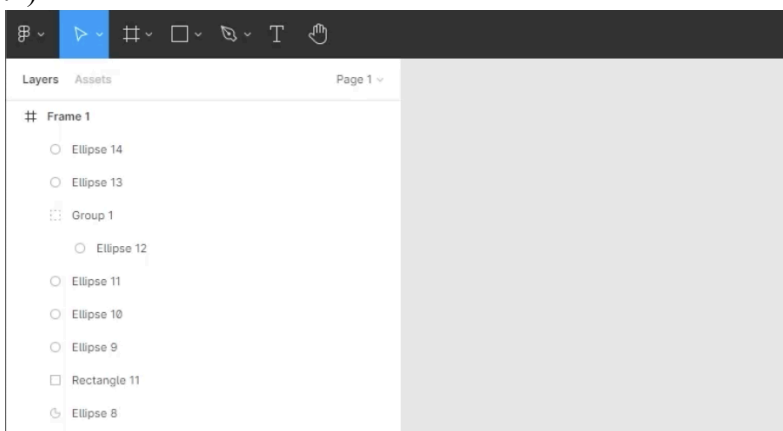


Рис. 13.1 – Фрейм під назвою «Frame 1» із вкладеними елементами

Фрейми можна вкладати один в інший для створення складної структури. Це дозволяє гнучко працювати з дизайном, групуючи елементи у логічні блоки, які легко редагуються. Для адаптивного розташування об'єктів у межах фрейму можна налаштувати

обмеження. Це дозволить контролювати, як елементи реагуватимуть на зміну розміру фрейму (фіксоване положення, прив'язка до сторін фрейму тощо).

Також фрейми підтримують функцію Auto Layout, що автоматизує процес розташування елементів у межах фрейму. Ця функція дозволяє задавати відстань між об'єктами, орієнтацію (горизонтально чи вертикально) та вирівнювання, що полегшує створення адаптивного і впорядкованого дизайну.

Група у Figma – це об'єднання кількох елементів дизайну в спільну логічну одиницю для спрощення роботи з ними. Групи дозволяють спільно переміщувати, масштабувати або редагувати об'єкти, не порушуючи їх відносного розташування один до одного. При цьому кожен елемент у групі зберігає свою індивідуальну структуру та властивості.

Групи є зручним інструментом для тимчасової організації елементів, що дозволяє дизайнерам працювати ефективніше при переміщенні або налаштуванні кількох об'єктів одночасно. При цьому вони не надають додаткових можливостей, таких як адаптивність чи обмеження, які притаманні фреймам. Групи застосовуються для об'єднання елементів, які потрібно одночасно редагувати, але які не вимагають додаткових налаштувань або поведінки. Наприклад, об'єднання тексту та іконки для створення кнопки, групування блоків при дизайні форм або об'єднання декількох елементів, щоб уникнути випадкового їх переміщення.

Створити групу можна, виділивши необхідні елементи та натиснувши **Ctrl + G (Windows)** та **Cmd + G (Mac)** або за допомогою контекстного меню. Для розгруповання використовується **Ctrl + Shift + G** або **Cmd + Shift + G**.

Хоча групи та фрейми у Figma на перший погляд виконують схожу функцію об'єднання елементів, вони мають суттєві відмінності у своїх можливостях та призначенні.

Фрейми є повноцінними контейнерами, які дозволяють організовувати дизайн з глибокою ієрархією. Вони підтримують такі функції, як обмеження (Constraints) для адаптивного розташування елементів, Auto Layout для автоматичної побудови структури та вкладені фрейми для створення складних макетів. Завдяки цим можливостям, фрейми використовуються для побудови структурованого та адаптивного інтерфейсу, який може змінюватися залежно від вимог дизайну.

Групи, натомість, є простішим інструментом для тимчасового об'єднання об'єктів. Вони не підтримують адаптивність і не впливають на поведінку елементів усередині них. Основна функція груп – це спільне редагування або переміщення об'єктів, зберігаючи їх відносне розташування. Проте масштабування групи впливає на всі елементи всередині неї, що може спотворити їхні пропорції, чого не відбувається у фреймах.

Шари (Layers) у Figma є ключовою частиною роботи з дизайном, оскільки вони визначають структуру всіх об'єктів на полотні та організовують їх у логічні блоки. Кожен елемент, створений у Figma, автоматично стає окремим шаром, що дозволяє легко управляти його властивостями, позиціонуванням і взаємодією з іншими елементами.

Основні види шарів:

– **Фрейми.** Фрейми є контейнерами, що об'єднують інші шари (текст, фігури, зображення або вкладені фрейми) у логічну структуру. Вони мають розширені можливості, такі як налаштування обмежень (Constraints), автоматичне розташування (Auto Layout) та адаптивність. Фрейми можуть служити як основою для дизайну екрану, так і елементами інтерфейсу.

– **Групи.** Групи об'єднують кілька елементів у єдиний шар для спрощення їх редагування чи переміщення. На відміну від фреймів, групи не мають функцій адаптивного дизайну й служать лише для тимчасової організації об'єктів.

– **Фігури.** Це базові графічні елементи, такі як прямокутники, кола, лінії та інші геометричні фігури. Шари фігур можуть редагуватися, змінювати колір, розміри та мати додаткові ефекти, як-от тіні, градієнти чи розмиття.

– **Текстові шари.** Текстові шари містять текстові елементи, що можуть бути стилізовані за допомогою шрифтів, розмірів, кольорів і вирівнювання. Вони є незамінними для роботи з інтерфейсами, де необхідно відобразити заголовки, кнопки чи описові блоки.

– **Зображення.** Ці шари використовуються для імпорту растрових графічних файлів (JPEG, PNG) або ілюстрацій, що інтегруються у макет. Зображення можуть бути змінені за розміром, обрізані або включені у фрейм як частина дизайну.

– **Компоненти.** Це спеціальні шари, що створюються для повторного використання елементів дизайну (наприклад, кнопки чи навігаційні блоки). Копії компонентів називаються інстансами (Instances), які можна редагувати з можливістю синхронізації з основним компонентом.

У Figma шари організовані у вигляді деревоподібної **ієрархії**, де кожен елемент знаходиться у певній структурі:

- Фрейми є основними контейнерами на верхньому рівні, які можуть містити вкладені шари.
- Всередині фреймів розташовуються групи та інші шари (текст, фігури, зображення).
- Групи об'єднують кілька елементів у єдиний шар, але не впливають на їхні властивості чи поведінку.
- Вкладені фрейми дозволяють створювати складні багаторівневі структури, зберігаючи порядок і зручність роботи.
- Компоненти та інстанси займають особливе місце у ієрархії, оскільки вони можуть повторюватися й синхронізуватися з основним елементом.

Маски у Figma дозволяють приховати частину об'єктів, не змінюючи їх структури. Вони використовуються для створення ефектів обрізання, відображення певних частин зображень чи інших елементів, що дозволяє дизайнерам маніпулювати вмістом без зміни самих об'єктів. Маска визначається певною формою або об'єктом, який використовується для обмеження видимості іншого елемента. У Figma маски працюють за принципом: об'єкт, який є маскою, визначає межі видимості для об'єкта, що знаходиться під нею. Все, що знаходиться за межами маски, стає прихованим.

Щоб створити маску, потрібно спочатку вибрати форму, яка буде використовуватись як маска. Після цього вибирається елемент або група елементів, які необхідно обрізати, і ці елементи розміщуються під маскою в ієрархії шарів. Потім обирається команда для застосування маски (Use as Mask), і елемент буде обрізано відповідно до форми маски.

Обмеження (Constraints) – це інструмент у Figma, який дозволяє контролювати, як об'єкти реагують на зміни розміру їхніх контейнерів (фреймів). Вони забезпечують адаптивність дизайну, дозволяючи елементам залишатися у визначеному положенні або змінювати свої пропорції залежно від розмірів фрейма. Це особливо важливо для створення інтерфейсів, що мають бути гнучкими та відповідати різним екранам або розмірам вікон. (Рис. 13.2)

Коли об'єкт знаходиться всередині фрейма, обмеження визначають його положення відносно сторін цього фрейма. Наприклад, якщо об'єкт закріплено зліва та зверху, він залишиться у тій самій позиції при зміні розміру фрейма. Якщо обмеження встановлено на "розтягування" (Left and Right), об'єкт змінюватиме ширину разом зі зміною розмірів фрейма.

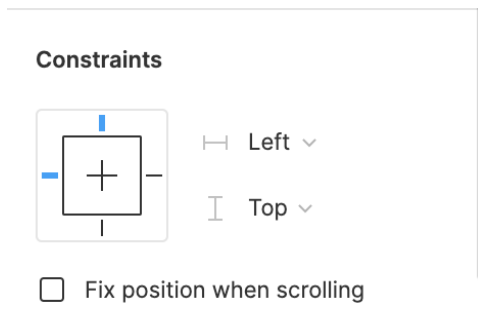


Рис. 13.2 – Інструмент Constraints

Завдання: Створити адаптивну до горизонтального масштабування картку користувача у програмі Figma. У роботі використати отримані знання із типів шарів, ієрархії та налаштування обмежень.

Хід роботи:

1. Переходимо у розділ Community у Figma, шукаємо плагін Feather Icons та встановлюємо його. (Рис. 13.3)

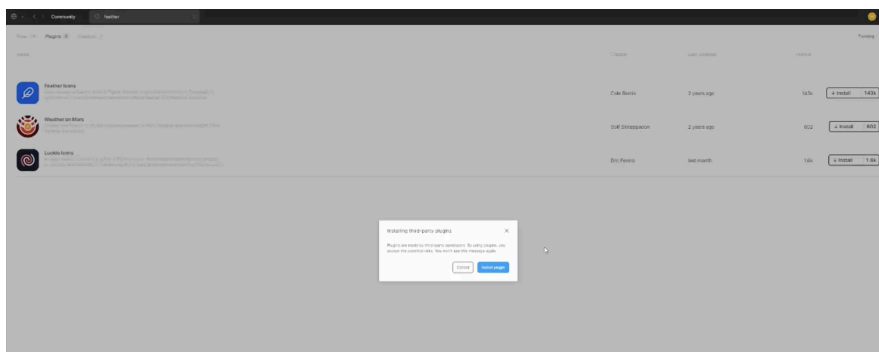


Рис. 13.3 – Вікно встановлення плагіну Feather Icons

2. Таким самим чином встановлюємо плагін User Profile.
3. Створюємо новий проєкт та фрейм у ньому. У фреймі будуємо заготовку під користувацьку картку за допомогою інструменту Rectangle і параметрів його налаштування. (Рис. 13.4)

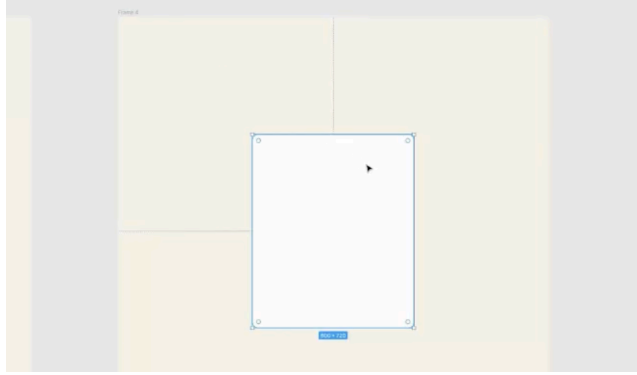


Рис. 13.4 – Етап створення картки користувача у Figma

4. Наповнюємо картку користувача додатковими елементами, такими як написи, фото профіля, іконки і рядок стану. Звертаємо увагу на структуру і ієрархію шарів. Використовуємо встановлені плагіни (запускаємо плагіни через контекстне меню) (Рис. 13.5)

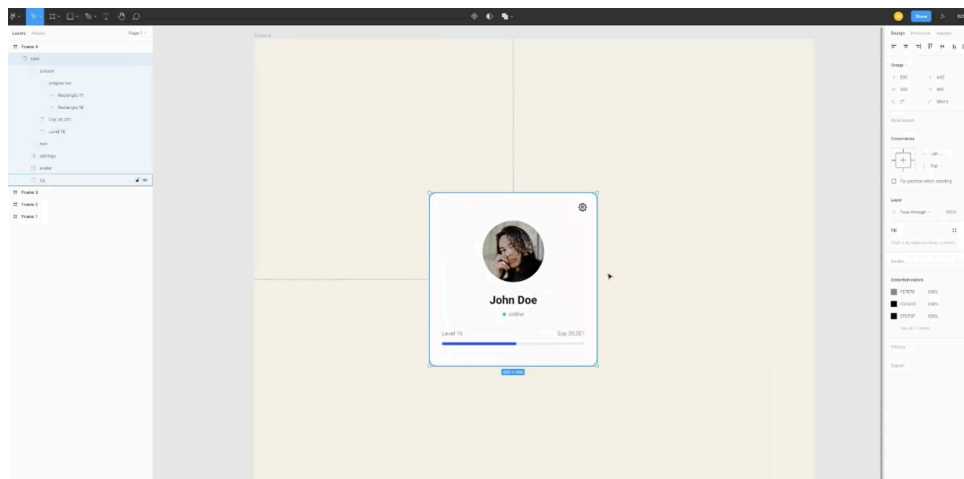


Рис. 13.5 – Картка користувача у Figma

5. За допомогою налаштування обмежень (Constraints) робимо картку користувача адаптивною до горизонтального масштабування. (Рис. 13.6)

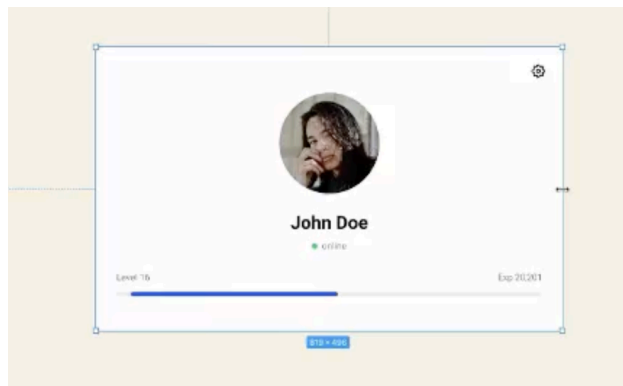


Рис. 13.6 – Адаптивна картка користувача у Figma

6. Зробіть висновок по роботі.

Домашнє завдання:

Представте власні приклади створення і налаштування фреймів та груп об'єктів у середовищі Figma. Продемонструйте використання масок для об'єктів і їхнє налаштування, а також роботу з системою шарів.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Опишіть, що таке фрейм і як його створити у середовищі Figma.
2. Розкрийте, які основні відмінності груп об'єктів та фреймів.
3. Поясніть, який принцип ієрархії шарів використовується у Figma.
4. Згадайте, які основні параметри обмежень (Constraints) передбачені у Figma.
5. Вкажіть, яким чином здійснюється групування об'єктів у Figma.
6. Подумайте, яким чином налаштувати базові ефекти шарів.

Література: [4] с. 46–48, 62–65.

ЛАБОРАТОРНА РОБОТА № 14.

Тема: Робота з текстом у середовищі Figma.

Мета роботи: Вміти створювати текстові об'єкти та налаштовувати їхній стиль засобами сервісу Figma.

Короткі теоретичні відомості

Типографіка – це техніка організації тексту для досягнення ефективної комунікації та естетичної привабливості. Вона охоплює вибір шрифтів, розмірів, інтервалів, вирівнювання та інших аспектів оформлення тексту, що забезпечують його зручність для сприйняття та відповідність контексту. Типографіка є невід'ємною частиною дизайну та відіграє важливу роль у створенні читабельних та привабливих текстових елементів.

Figma надає широкі можливості для роботи з текстовими елементами, що дозволяє створювати естетично привабливі й читабельні макети. У Figma можна налаштовувати всі основні параметри тексту, такі як шрифт, розмір, інтервал, вирівнювання та інші, що забезпечує повний контроль над типографікою проєкту.

Інструмент Text є основним інструментом для створення текстових елементів у Figma. Для його активації достатньо натиснути на значок "T" у верхньому меню або використувати клавішу "T" на клавіатурі. Після активації інструмента можна клікнути на робочу область та почати вводити текст. (Рис. 14.1)

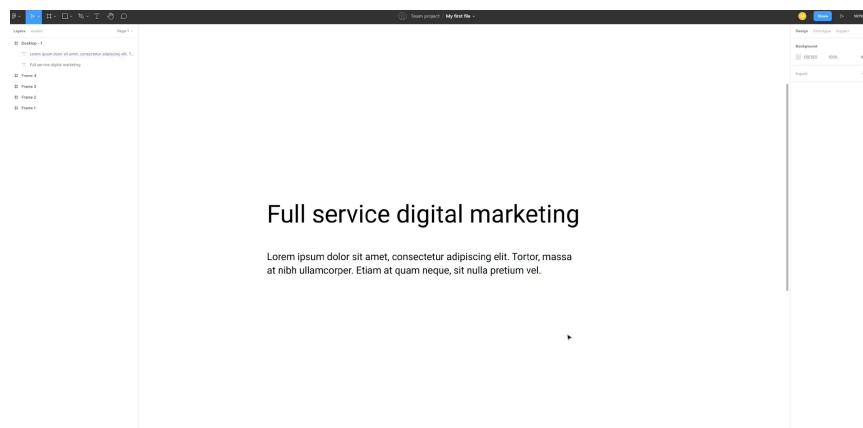


Рис. 14.1 – Приклад використання інструменту Text

У Figma доступно багато вбудованих шрифтів, а також є можливість підключати зовнішні шрифти через Google Fonts або через інтерфейс з іншими додатками для шрифтів. Для вибору шрифту достатньо клікнути на відповідний випадаючий список у панелі властивостей і обрати потрібний шрифт. Шрифти можуть бути різними за стилем (Regular, Bold, Italic) та вагою.

Також у Figma можна легко налаштувати **розмір шрифту** (Font Size) через панель властивостей. Це дозволяє точно регулювати розмір тексту залежно від його призначення та контексту на макеті. Розмір можна задавати в пікселях (px), що забезпечує більшу точність.

Інтерліньяж (Line Height) визначає відстань між рядками тексту. У Figma даний параметр можна налаштувати через панель властивостей. Правильний вибір інтерліньяжу покращує сприйняття тексту, роблячи його більш читабельним та комфортним для очей.

Літерний інтервал (Letter Spacing) – це відстань між окремими літерами в тексті. Цей параметр допомагає зробити текст більш чітким або, навпаки, більш стиснутим, залежно від потреби дизайну.

В Figma доступні різні варіанти **вирівнювання тексту** (Text Alignment) – ліве, праве, по центру та по ширині. Цей інструмент дозволяє коригувати текст для досягнення бажаного вигляду та узгодженості з іншими елементами на сторінці.

Також є можливість змінювати **ширину текстового поля** (Text Box Width), щоб адаптувати текст до необхідного розміру або області. Це дозволяє створювати адаптивні тексти, що коректно відображаються в межах встановлених розмірів контейнера.

Всі налаштування для текстових шарів знаходяться праворуч у відповідній панелі, що включає в себе великий набір можливостей для форматування тексту. (Рис. 14.2)

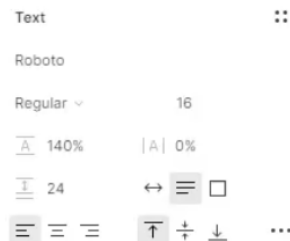


Рис. 14.2 – Блок параметрів налаштування тексту

Основні параметри налаштування тексту на панелі «Text»:

- Font Family – вибір шрифту для тексту.
- Font Style – стиль шрифту (Regular, Bold, Italic).
- Font Size – розмір шрифту в пікселях.
- Line Height – вертикальний інтервал між рядками.
- Letter Spacing – відстань між літерами.
- Text Alignment – вирівнювання тексту (ліве, праве, по центру, по ширині).
- Text Color (Fill) – колір тексту.
- Font Weight – товщина шрифту.
- Italic – перетворення тексту на курсив.

- Text Case – зміна регістру тексту (UPPERCASE, lowercase).
- Paragraph Spacing – відстань між абзацами тексту.

Правила типографіки у веб-дизайні:

- Читабельність та зручність сприйняття тексту.

Читабельність є однією з основних вимог до тексту в веб-дизайні. Текст має бути легким для сприйняття, що досягається шляхом правильного вибору шрифтів, розмірів, міжрядкових інтервалів та іншого налаштування типографії. Вибір шрифту повинен забезпечувати його чіткість і комфортне читання, особливо на екранах різного розміру. Наприклад, для основного тексту на веб-сторінках зазвичай використовуються шрифти без засічок, які покращують видимість та читабельність при малих розмірах. Розмір шрифту для основного тексту не повинен бути меншим за 16px, щоб забезпечити зручність читання на різних пристроях.

- Ієрархія тексту та його структурованість.

Ієрархія тексту є важливою складовою організації інформації на веб-сторінці. Вона дозволяє користувачам швидко орієнтуватися і знаходити необхідну інформацію, а також зрозуміти її важливість. Чітке виділення заголовків, підзаголовків і основного тексту за допомогою різних шрифтів, стилів, розмірів та ваги шрифтів допомагає створити візуальну структуру, яка полегшує сприйняття контенту. Заголовки та підзаголовки повинні бути помітними та контрастними, в той час як основний текст має залишатися комфортним для читання, не відволікаючи увагу від основних ідей.

- Консистентність та використання стилів.

Консистентність у використанні типографічних елементів на всіх сторінках веб-сайту є важливим аспектом у створенні зручного інтерфейсу. Використання однакових шрифтів, стилів та налаштувань типографії на різних сторінках забезпечує узгодженість дизайну, що допомагає користувачам легше орієнтуватися в контенті. У веб-дизайні доцільно використовувати визначені стилі для заголовків, підзаголовків і основного тексту, а також для інших елементів, як-от списків або цитат. Це не тільки зберігає єдність оформлення, але й полегшує редагування тексту, дозволяючи змінювати стиль тексту в одному місці, що автоматично оновлюється на всіх сторінках.

- Контрастність та кольорова палітра.

Контраст між текстом і фоном є ключовим фактором у забезпеченні зручності читання. Недостатній контраст може призвести до труднощів у сприйнятті тексту, що негативно позначається на користувацькому досвіді. Веб-дизайнери повинні обирати кольори тексту та фону так, щоб вони були достатньо контрастними для легкого сприйняття навіть при слабкому освітленні або на екранах з низькою роздільною здатністю. Оскільки кольори можуть впливати на емоційне сприйняття інформації, важливо використовувати кольорову палітру, яка відповідає бренду та дизайну сайту.

- Адаптивність типографіки.

Адаптивність тексту є важливою складовою у сучасному веб-дизайні, оскільки користувачі звертаються до веб-ресурсів з різних пристроїв (настільних комп'ютерів, планшетів, мобільних телефонів). Текст на веб-сторінках повинен адаптуватися до різних розмірів екрану та забезпечувати зручність читання на будь-якому пристрої. Окрім того, адаптивність типографії включає в себе оптимізацію інтервалів між рядками, міжлітерного інтервалу та розмірів шрифтів, що змінюються в залежності від розміру екрану.

- Мініمالізм та чіткість в оформленні тексту.

Мінімалізм є важливою складовою сучасного веб-дизайну. Занадто багато тексту чи надмірні декоративні шрифти можуть перевантажити сторінку та знизити зручність навігації. Дизайнери повинні забезпечити чіткість та лаконічність текстових елементів, щоб користувачі могли швидко знайти необхідну інформацію без зайвих зусиль. Слід уникати використання великої кількості різноманітних шрифтів або занадто складних шрифтів, оскільки це може зробити дизайн перевантаженим і менш естетичним. Простота у виборі шрифтів і стилів дозволяє тексту бути більш зрозумілим для користувача.

Завдання: Створити текстову композицію у Figma, використовуючи інструменти роботи з типографікою та правила типографіки у веб-дизайні.

Хід роботи

1. Створіть новий проєкт та фрейм для роботи.
2. Створіть композицію у вигляді головного екрану веб-сайту (як приклад). Для цього оберіть задній фон, розмістіть текстові блоки згідно логіки веб-дизайну.
3. Підберіть шрифти, налаштуйте параметри текстових блоків.
4. Перевірте дизайн на відповідність основним принципам типографіки. Також, за бажанням, додайте оформлення (графічні об'єкти, рисунки тощо) (Рис. 14.3)

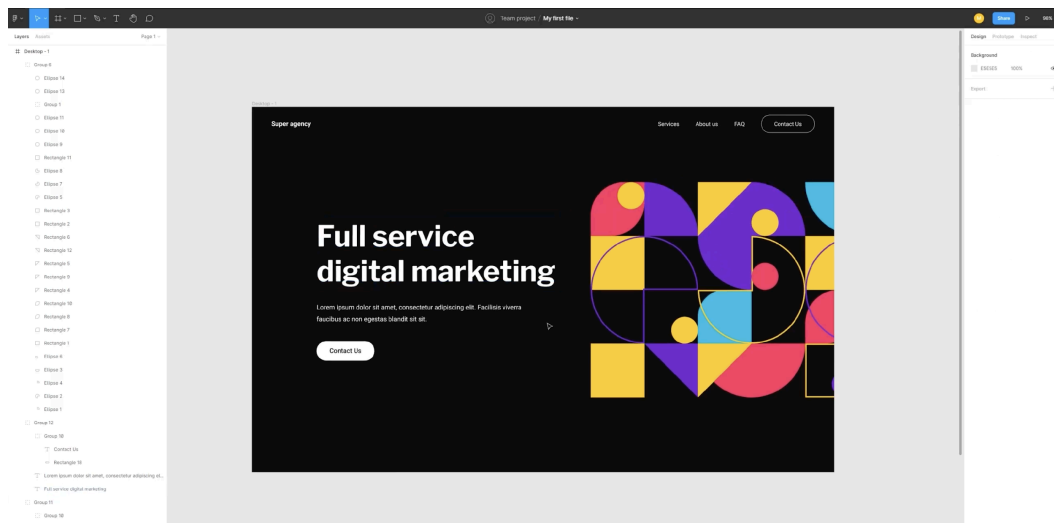


Рис. 14.3 – Приклад текстової композиції

5. Зробіть висновок по роботі.

Домашнє завдання:

Створіть та відредагуйте довільні текстові об'єкти, налаштуйте стиль тексту, шрифтів і розмірів. Застосовуйте правила та принципи типографіки у веб-дизайні.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами

(знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Дайте визначення поняттю типографіки у веб-дизайні.
2. Згадайте правила типографіки у веб-дизайні.
3. Поясніть, як створюються текстові об'єкти у Figma.
4. Опишіть, які основні інструменти роботи із текстом існують в програмному середовищі Figma.

Література: [3] с. 161–167; [4] с. 27–31.

ЛАБОРАТОРНА РОБОТА № 15.

Тема: Встановлення стилів у середовищі Figma.

Мета роботи: Набуття навичок створення та налаштування власних стилів у середовищі Figma, вивчення основних параметрів стилізації базових ефектів шарів.

Короткі теоретичні відомості

Стилі у Figma є важливим інструментом для систематизації й уніфікації дизайну, що дозволяє спростити роботу над проєктами, забезпечити їх візуальну послідовність та підвищити продуктивність роботи. Стилі представляють собою заздалегідь збережені налаштування певних властивостей, таких як кольори, текстові параметри, ефекти або сітки, які можна повторно використовувати у проєкті. Вони дозволяють швидко застосовувати однакові параметри до різних об'єктів, що особливо актуально при проєктуванні великих та складних інтерфейсів.

Основна функція стилів полягає у стандартизації дизайну. Коли йде робота над дизайном проєкту, який містить багато елементів, важливо зберігати єдиний візуальний підхід. Стилі дозволяють створювати збережені пресети (налаштування), які можна легко застосовувати або оновлювати. Це в значній мірі мінімізує необхідність ручного налаштування кожного елемента, скорочуючи час на правки та підтримку узгодженості.

У Figma виділяються наступні типи стилів:

- **Колірні стилі** дозволяють зберігати палітру кольорів, яка використовується у проєкті, для подальшого їхнього швидкого застосування до фреймів, шарів, тексту, обводок чи заливок. Це зручно для роботи з брендovими кольорами або для підтримки дизайну в єдиній кольоровій гамі. (Рис. 15.1)

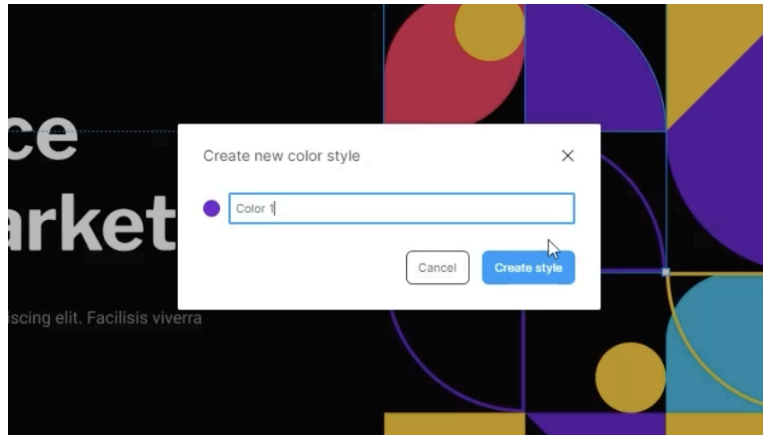


Рис. 15.1 – Вікно створення стилю кольору

- **Текстові стилі** відповідають за збереження параметрів типографіки, таких як шрифт, розмір, товщина, міжрядковий інтервал та вирівнювання. Використання текстових стилів допомагає структуровано організувати заголовки, підзаголовки, основний текст та інші текстові елементи в інтерфейсі.
- **Ефектами** у Figma можна зберігати параметри таких властивостей, як тіні (Drop Shadow), розмиття (Blur), внутрішнє світіння (Inner Shadow) та інші подібні ефекти.
- **Стилі сіток** дозволяють зберігати параметри модульних сіток, що використовуються для вирівнювання об'єктів і створення структурованого макету. Це допомагає при створенні адаптивного дизайну, де макети мають підлаштовуватися під різні розміри екранів.

Для створення нового стилю необхідно налаштувати потрібні параметри об'єкта (наприклад, колір, ефекти чи типографіку), після чого зберегти їх як стиль. Весь процес створення та редагування стилю відбувається з панелі налаштувань, праворуч від основного робочого простору.

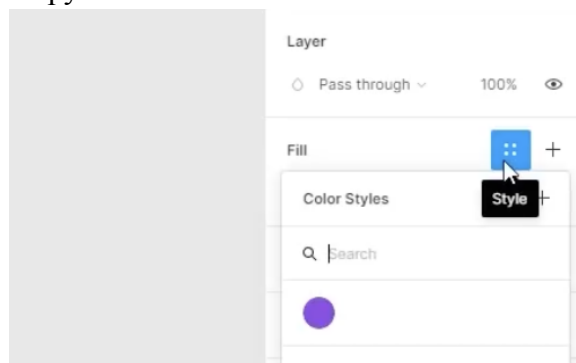


Рис. 15.2 – Панель створення нового стилю кольору

Ефекти шарів у Figma є базовими інструментами для стилізації об'єктів. До основних параметрів ефектів належать заливка (Fill), обведення (Stroke) та ефекти панелі Effects, які дозволяють створювати тіні, розмиття й інші візуальні властивості. Для цих параметрів також можна зберігати стилі, що забезпечує узгодженість і швидкість у роботі над проєктами.

Fill – це параметр, що відповідає за заповнення внутрішньої області об'єкта кольором, градієнтом або іншим візуальним ефектом. Заливка є основним способом роботи з кольорами у Figma, і її можна застосовувати до будь-якого об'єкта: тексту, фреймів, геометричних фігур або компонентів.

Типи заливки:

- Суцільний колір (Solid): застосування одного кольору з можливістю налаштування прозорості (Opacity).
- Градієнти (Gradients): лінійний (Linear), радіальний (Radial), кутовий (Angular) та градієнт діаметра (Diamond). Градієнти дозволяють плавно змінювати кольори в межах об'єкта. (Рис. 15.3)

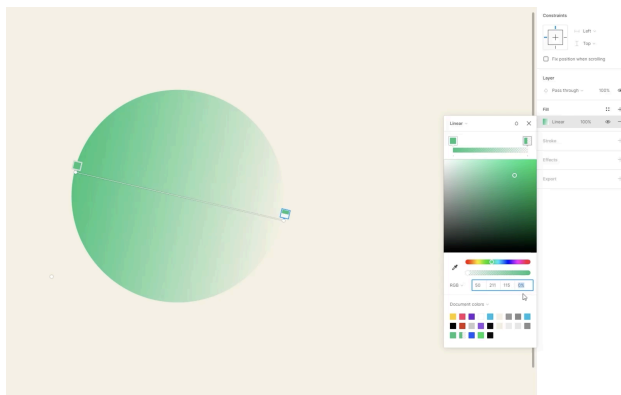


Рис. 15.3 – Створення ефекту заливки «Градієнт»

- Зображення (Image Fill): можливість використовувати зображення як текстуру або фоновий елемент.

Також для кожного шару заливки можна окремо налаштовувати рівень прозорості. До того ж Figma дозволяє додавати кілька заливок до одного об'єкта, що дає можливість створювати складні візуальні ефекти.

Stroke відповідає за зовнішнє або внутрішнє обведення об'єкта лінією. У налаштуваннях даного ефекту є наступні параметри:

- Товщина лінії (Weight): визначає ширину обведення, яку можна налаштовувати залежно від потреб.
- Розташування лінії: обведення може бути розташоване з внутрішнього боку (Inside), зовнішнього боку (Outside) або по центру контуру (Center).
- Стили ліній: Figma дозволяє застосовувати різні стилі обведення, включаючи суцільні лінії, пунктир (Dashed) або змішані патерни.
- Кінці та кути: для обведення можна налаштовувати типи кутів (згладжені, скошені) та кінців ліній (круглі, квадратні).
- Колір і прозорість: обведення можна налаштувати аналогічно заливці, включаючи застосування кольорів, градієнтів або текстур.

Панель Effects у Figma містить інструменти для створення допоміжних візуальних ефектів, які додають глибини, текстурності та акцентів до об'єктів. До основних ефектів належать:

1. Тіні (Drop Shadow):

Тіні використовуються для створення ефекту об'єму та виділення об'єкта на тлі. Параметри тіні включають:

- X i Y Offset – зміщення тіні по горизонталі та вертикалі.
- Blur – ступінь розмиття тіні.
- Spread – розширення тіні для збільшення її площі.
- Color – налаштування кольору та прозорості тіні.

2. Внутрішні тіні (Inner Shadow):

Внутрішні тіні додають тіні всередині об'єкта, створюючи ефект заглиблення або внутрішнього об'єму. Параметри аналогічні до зовнішніх тіней.

3. Розмиття (Layer Blur та Background Blur):

- Layer Blur: застосовує розмиття до всього об'єкта, створюючи ефект «сфокусованості» на інших елементах.
- Background Blur: розмиває фон об'єкта, що дозволяє створювати ефект скляних панелей або напівпрозорих елементів.

Ефекти, налаштовані на панелі Effects, можна також зберігати як **стили**. Це дозволяє швидко застосовувати однакові параметри тіней або розмиття до різних об'єктів у проєкті. У разі внесення змін до стилю всі об'єкти, що його використовують, будуть автоматично оновлені.

Завдання: Створити графічну композицію з шести геометричних фігур з використанням різних параметрів налаштування ефектів та застосуванням стилів.

Хід роботи

1. Створюємо новий проєкт та фрейм для роботи.
2. За допомогою інструментів графічних фігур створюємо об'єкти для подальшого накладання ефектів і стилів.
3. Створюємо власні стилі, використовуючи набір налаштувань різних параметрів – заливки, тіні, розмиття тощо. Зберігаємо створені стилі.
4. Застосовуємо стилі до графічних об'єктів.

Приклад з чотирьох об'єктів з різними стилями та панель стилів представлено на рис. 15.4.

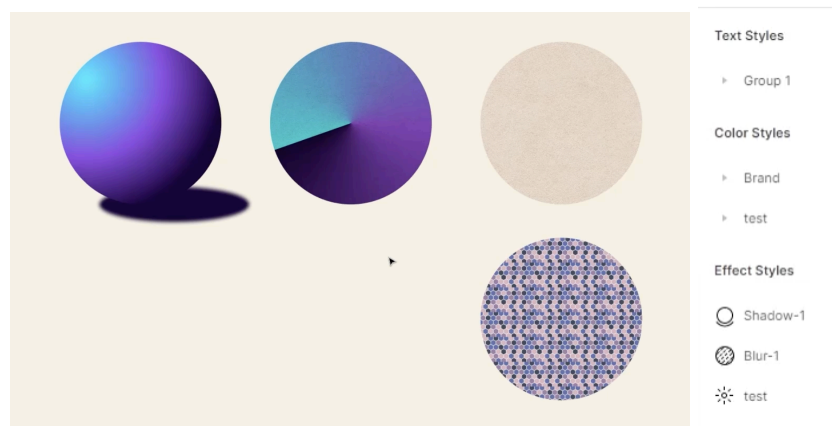


Рис. 15.4 – Приклад оформлення графічної композиції зі стилями

5. Зробіть висновок по роботі.

Домашнє завдання:

Створіть іншу групу графічних об'єктів. Застосуйте до них стилі із лінійною, радіальною, кутовою градієнтною заливкою, а також градієнт діаметра.

Рекомендації щодо оформлення звіту:

Звіт з лабораторної роботи повинен включати порядковий номер лабораторної роботи та її повну назву, дату виконання, формулювання мети роботи, стисло викладені теоретичні відомості по темі роботи, перелік виконаних завдань до роботи зі скриншотами (знімками екрана), що демонструють процес виконання. Звіт завершується висновком до виконаної лабораторної роботи.

Контрольні питання

1. Визначте, як встановлюються налаштування стилю у Figma.
2. Поясніть, яким чином налаштувати базові ефекти шарів.
3. Назвіть, які параметри можна зберегти як стилі у Figma.
4. Представте способи встановлення кольорів і налаштування відтінків у Figma.
5. Опишіть, що можна налаштувати на панелі Effects у Figma.

Література: [3] с. 147–161; [4] 118–123.

СПИСОК ЛІТЕРАТУРИ

Основна література:

1. Бородкіна І.Л., Бородкін Г.О. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів. – Київ: Ліра-К, 2020. – 212 с. (28 примірників)
2. Пасічник В.В., Пасічник О.В., Угрин Д.І. Веб-технології. – Львів: Видавництво "Магнолія", 2024. – 336 с. (11 примірників)
3. Мельник Р.А. Програмування веб-застосунків (фронт-енд та бек-енд). – Львів: Видавництво Львівської політехніки, 2018. – 247 с. (19 примірників)
4. Стаяно Ф. FIGMA. Проектування і прототипування інтерфейсів відповідно до принципів UX/UI. – Харків: Фоліо, 2021. – 370 с. (7 примірників)
5. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну: [Навч. посіб.]. – К.: Вид. група ВНУ, 2009. – 336 с. (16 примірників)

Додаткова література:

6. Емброуз Г., Леонард Н. Основи графічного дизайну 02. Дизайнерське дослідження. Пошук успішних креативних рішень. – Пер. з англ. – К.: ArtHuss, 2019. – 192 с.

7. Moore S., Wilkins A. Strategic Web Design & e-Commerce. – Mujo Learning Systems, 2021. – 385 p.
8. McGrath M. HTML, CSS & JavaScript in easy steps. – UK: In Easy Steps Limited, 2020. – 480 p.