**Відповіді на запитання**

Макет веб-інтерфейсу - це вигляд веб-інтерфейсу з усіма деталями (всі елементи, шрифти,кольори, іконки, зображення та інше), який створюється в графічному редакторі.

Графічний редактор - це програмне забезпечення, яке призначене для роботи із ком’ютерною графікою.

Види комп’ютерної графіки. Для створення дизайну веб-інтерфейсів найкраще використовувати растрову та векторну графіки.

Adobe Photoshop - це графічний редактор компанії Adobe Systems, який ідеально підходить для роботи з растровою графікою та для створення складного дизайну веб-сторінок. На сьогоднішній день його використовують все менше і менше. Причиною цього є те, що він має складний інтерфейс, є дуже громіздким і платним. Працює лише на Windows та Mac OS.

Adobe Experience Design - це ще один графічний редактор компанії Adobe System, який призначений для створення дизайну графічних інтерфейсів.

Figma - графічний редактор векторної графік, який призначений для створення дизайну графічних інтерфейсів. Він має веб-версію, що дозволяє його використовувати на будь-якій операційній системі. А, також, він дозволяє кільком дизайнерам працювати над одним макетом одночасно.

Sketch - векторний графічний редактор для операційної системи Mac OS. Він використовується для проектування інтерфейсів мобільних додатків та веб-сайтів. Підтримує можливість створення інтерактивних прототипів.

**HTTP** – це протокол прикладного рівня, розроблений для обміну гіпертекстовою

інформацією в мережі Internet. Протокол використовується в одному з найпопулярніших

ресурсів мережі Internet – Word Wide Web – з 1990 року.

Одним з інструментів створення WEB-сторінок є мова HTML. Крім того на даний

момент є досить перспективна мова XML. Основним елементом HTML є дескриптор (tag).

Тег – це основний елемент кодування, прийнятий в стандарті HTML. В документі

HTML все залежить від дескрипторів. Дескриптори визначають правила розмітки в

документі. Всі дескриптори беруться в кутові дужки < >. Наприклад дескриптор розриву

абзацу <P> (Page), дескриптор <HR> - розмітка горизонтальною лінією (horizontal role). Всі

дескриптори мають певні властивості, які називаються атрибутами. Атрибут – управляючий

елемент дескриптора, відповідно розміщується всередині кутових дужок дескриптора.

**Існує два типи дескрипторів.**

***1***.Контейнерні – це дескрипторна пара, яка складається з початкового

(відкриваючого) і кінцевого (закриваючого) дескрипторів а також вмісту контейнеру над

яким проводиться операція. Запис контейнерного дескриптора:

<TAG> вміст контейнеру </TAG>

де, TAG – ім’я реального контейнерного дескриптора.

     Таким чином, контейнерний дескриптор виконує задачу розмітки над вмістом

контейнеру, наприклад:

<PRE> Це відформатований текст </PRE>

***2***.Одиночний дескриптор, або пустий не містить інформації і виконує в основному

самостійну задачу. Наприклад, дескриптор <HR> створює горизонтальну лінію в HTML

документі.

      Отже – елементи HTML-документу – все те, що знаходиться між початковим і

кінцевим дескрипторами а також самі дескриптори.

Елементами документу можуть бути – зображення; фрагменти тексту; форми;

таблиці; списки; посилання; текстове поле; кнопки; заголовок документу; основне тіло

документу.

<B>...</B>: Дескриптор для встановлення напівжирного напису тексту.

<I>...</I>: Дескриптор для встановлення курсиву в тексті.

<U>...</U>: Дескриптор для підкреслення тексту.

<TT>...</TT>: Дескриптор моношириного шрифту.

<SMALL>...</SMALL>: Дескриптор, що зменшує шрифт.

<BIG>...</BIG>: Дескриптор, що збільшує шрифт.

<SUP>...</SUP>: Дескриптор, що робить надстрочний шрифт.

<SUB>...</SUB>: Дескриптор, що робить підстрочний шрифт.

<STRIKE>...</STRIKE>: Дескриптор, що робить перекреслений шрифт.

В тих місцях де стоїть три крапка (...) пишеться текст, що ви його форматуєте.

Дескриптори логічних стилів вказують на характер виділення, а не на спосіб його

відображення на сторінці. Браузер на дескриптор логічних стилів реагує таким чином, що

сам визначає як відформатувати текст найбільш оптимально по відношенню до іншої

частини тексту Web сторінки. Всі дескриптори логічних стилів є контейнерні.

<EM>...</EM> - виділений текст, відмінність від <I> - браузер сам визначає стиль

виділення тексту.

<STRONG>...<STRONG>: сильно виділений текст.

<CITE>...</CITE>: текст у вигляді цитати.

<CODE>...</CODE>: текст у вигляді фрагменту коду HTML.

<DEN>...</DEN>: текст у вигляді визначення.

<SAMP>...</SAMP>: текст у вигляді фрагменту коду (аналогічно стилю <CODE>).

<KBN>...</KBN>: текст у вигляді назви клавіші клавіатури.

<VAR>...</VAR>: текст, котрий визначає перемінну або значення.

<ACRONYM>...</ACRONYM>: абревіатура (акронім) і його розшифрування.

Приклад:

<ACRONYM TITLE=”Оперативний запам’ятовуючий пристрій”>OЗУ</ACRONYM>

На екрані браузера будемо мати:

OЗУ

Навігація по вузлі - це той механізм, що дозволяє відвідувачу знайти те, що ви

хочете йому показати. Від простоти і зрозумілості системи навігації по вузлі, продуманості її

елементів залежить, чи досягне відвідувач своєї мети чи покине, так і не розібравшись в

нетрях гіпер-документів. Саме зручна навігація, а не красиві картинки, змушує відвідувачів

повертатися на ваш вузол, тому що вони знають, що з легкістю знайдуть необхідний

матеріал.

Навігація починається з домашньої сторінки вузла і наскрізь пронизує всю

структуру вузла, забезпечуючи зручний та швидкий інформаційний “серфінг” користувачу.

**Елементи навігації.**

Вся сутність WEB полягає у зв’язку. Якщо б не було цієї можливості саме 6 поняття

мережі зводилось би до публікації документів в Internet. Пов’язування ж дозволяє вийти за

межі документу і отримати доступ не тільки до інших документів, що відносяться до нього,

але й до інших cсилок Internet.

Спочатку цей метод зв’язку називався гіперзв’язком технічний прийом зв’язку з іншими

документами. На сьогоднішній день в WEB використовують зв’язки не тільки, як

гіперзв’язки, а й зв’язок з великою кількістю об’єктів і засобів мультимедіа. Цей новий вид

зв’язку називається - гіпермедіа.

Існує три основних види гіпертекстових звязків:

----- абсолютні зв’язки;

----- відносні зв’язки;

----- локальні зв’язки.

Для зв’язку в нас служить дескриптор якоря &amp;lt;A&amp;gt;...&amp;lt;/A&amp;gt; з атрибутом href. Вся

інформація, що записана в дескрипторі якоря &amp;lt;A&amp;gt; тут відображено зв’язок &amp;lt;/A&amp;gt;.

Абсолютний зв’язок використовує повну URL адресу. Це означає, що ви використали

абсолютно всю WEB-адресу, а не її частину. В зв’язку включена інформація про протокол і

домен. Ці дані дозволяють перйти на початкову сторінку по замовчуванню цієї WEB-

сторінки. Абсолютні ссилки важливі при задані в елементах прив’язки адрес не на вашому

вузлі, а на інших вузлах, що знаходяться на інших серверах.

Розміщення на WEB-сторінці зображень є першим кроком від простого форматування документів в світ дизайну. Зображення (малюнки, flech, video) надають HTML-сторінкам індивідуальність, колір, форму і зовнішній вигляд. Вони є потужним і важливим елементом WEB. Розміщення зображення на сторінці не підлягають якимось стандартам, а підпорядковуються певним угодам. Є декілька способів це зробити. Наприклад можна використати звичайні дескриптори, таблиці, і на кінець у відповідності зі стандартом HTML 4.0 зображення можна розміщати з допомогою листів стилів.

Для розміщення зображення на HTML-сторінки використовується дескриптор <IMG>. Він працює самостійно і йому не потрібний закриваючий дескриптор. Для дескриптора <IMG> потрібно вказати URL в атрибуті src.

Приклад:

<IMG src="1\mob1.jpg">

В вікні браузера будемо мати відносний зв’язок на початок:

Цієї стрічки достатньо, щоб розмістити рисунок на сторінці, якщо він знаходиться в тому ж місці, що і HTML-сторінка. Як правило розробники WEB-сторінок зберігають свої малюнки в якомусь одному каталозі і дають назви images (зображення) або graphics (графіка), в залежності від того яке слово розробнику більше подобається. Деякі розробники створюють лінки на зображення, що находяться за межами WEB-вузлу з використанням абсолютної URL. Це не рекомендується робити по кільком причинам: можливе погане з’єднання з іншим сервером, і малюнок може зовсім не завантажитись. Зображення завжди розміщають в середині розділу дескриптора <BODY>.

В дескриптора <IMG> є ряд атрибутів для керування зображенням (управління поведінкою, місцем розташування на сторінці, і зовнішнім видом зображення на сторінці).

В дескриптора <IMG> є ряд таких атрибутів:

src="x" - джерело зображення (її абсолютний або відносний URL);

width="x" - дозволяє браузеру наперед встановити ширину зображення;

height="x" - це є атрибут висоти і використовують разом з атрибутом width, браузер наперед встановлює місце для зображення на сторінці;

border="x" - атрибут, що визначає ширину рамки навколо малюнку;

align="x" - атрибут вирівнювання малюнку на сторінці по горизонталі і вертикалі;

alt="опис" - дозволяє текстовим браузерам виводити га екран замість малюнку його опис, цей опис відображається на екрані будь-якого браузера (не тільки текстового) до тих пір поки не відбудеться загрузка самого зображення. Крім того атрибут alt дозволяє виводити на екран елемент підказки з описом, коли вказівник мишки розміщений над зображенням.

hspace="x" - вільний простір по горизонталі; використовується для добавлення вільного місця справа і зліва від зображення. Встановлюється в вигляді числового значення.

vspace="x" - атрибут vspace такий самий, як атрибут hspace, але для вільного простору по вертикалі.

плаваючі зображення - плаваючі зображення, це обтікання тексту навколо зображення.

            Таблиці дозволяють більш ефективно розмістити на Web сторінці текст і

графіку. Вони використовуються також для представлення інформації у виді

логічної схеми, що значно спрощує її пошук. В HTML-таблицях може

міститись інформація любого виду. З їх допомогою можна створити чітку

структуру матеріалу на Web сторінці. Тому таблиці дуже часто

використовуються як інструмент макетування для якісного форматування

сторінки а також це практично єдиний засіб замальовки всієї таблиці або її

окремих частин кольором. Весь цифровий матеріал, поданий на сторінці, як

правило, може бути оформлений у вигляді таблиць. Таблиця складається з

рядків і стовпців. Кожна таблиця (якщо вона подана в явному виді, а не як

інструмент форматування) повинна мати назву, яку розміщують над таблицею і

друкують симетрично до тексту. Назву і слово "Таблиця" починають з великої

літери. Назву не підкреслюють. Заголовок граф повинні починатися з великих

літер, підзаголовки - з маленьких, якщо вони складають одне речення із

заголовком, і з великих, якщо вони є самостійними. Висота рядків повинна бути

не меншою 8 мм. Графу з порядковими номерами рядків до таблиці включати

не треба.

 Код таблиці в HTML міститься в контейнері ***<TABLE>.*** Дескриптори

***<TR>...</TR>*** описують один рядок, а за допомогою дескрипторів ***<TD> -***

комірку таблиці з даними та ***<TH>*** - комірка заголовку таблиці. Максимальна

кількість комірок в таблиці не може перевищувати добутку кількості рядків на

кількість стовпців. Закриваючі дескриптори </TD> та </TH> можуть не

використовуватись якщо є в кінці опису рядка є закриваючий дескриптор

</TR>. Всі рядки повинні містити однакову кількість комірок. Виключенням

може бути об’єднання за допомогою атрибуту COLSPAN=? декількох комірок

сусідніх стовпців в одну, а за допомогою атрибуту ROWSPAN=? – декількох

комірок сусідніх рядків (де ? – кількість об’єднаних елементів). Ці атрибути

описуються всередині дескрипторів <TD> та <TH>.

Робота з фреймами.

**Фрейми** — вічне джерело захоплення та страждань як для Web-дизайнерів, так і

для відвідувачів вузлів. Фрейми відбирають і без того малу площу екрану для

свого обрамлення. У зв'язку з цим, а також з огляду на відвідувачів, що має

обмеження по зору, їхнє використання вимагає акуратного програмування.

             Крім того, застосування фреймів змушує дизайнера писати більше коду. Тому

фрейми, як частина своїх розробок використовують тільки досвідчені

дизайнери — та й то нечасто, побоюючись відлякати відвідувачів вузла.

З іншого боку, застосування фреймів дозволяє дизайнеру обновляти тільки

частина сторінки, фіксуючи, наприклад, в одному фреймі систему навігації по

вузлі і залишивши другий фрейм для показу викликаної сторінки. Але саме

цікаве полягає в тому, що фрейми, особливо без обрамлення, дозволяють

дизайнеру створити сіткову систему на сторінці. Тим самим реальне

застосування фреймів йде від свого первісного призначення —

використовуватися винятково як інструмент для форматування сторінки і

гнучкого керування нею. У цьому фрейми подібні з таблицями (правда, слід

зазначити, що використання таблиць можливо тільки в межах сторінки).

              Web-дизайнер повинен добре уявляти собі призначення фреймів. Чи доводиться

застосування фреймів до створення розкреслених сторінок із привабливим чи

інтерфейсом складних макетів без рамок — у будь-якому випадку в руках

конструктора є могутній інструмент.

          Форма (form) – перевірений часом гнучкий метод взаємодії користувачів з Web-

вузлом. Це один з видів інтерфейсу взаємодії з аудиторією. На відміну від

статичних методів HTML, форма зв’язує не тільки користувачів і сторінки

вузла, але і сценарії, що виконуються на WEB-сервері. Відношення між

формою і програмою, що обробляє її дані регулюються стандартом

CGI (Common Gateway Interface – інтерфейс спільного доступу). CGI працює в

якості каналу, по котрому передається інформація із форми процесу , що її

обробляє. Форми можуть виконувати широкий ряд задач – від звичайного

отримання даних про користувачів вузла (його ім’я, адресу та ін.) до створення

ігр. CGI не єдина сучасна технологія, призначена для обробки даних форми.

В будь-якій формі існують дескриптори і деякі спеціальні елементи, які

називаються елементами управління. Існує декілька ключових елементів форм і

ряд пов’язаних з ними атрибутів котрі необхідно знати для створення форми.

<FORM>...</FORM> - це фундаментальний елемент всіх форм, дескриптори,

відкриваючі і закриваючі форму. Форма може мати декілька атрибутів. Основні

з них – action, котрий в якості значення приймає URL для передачі на вузол

інформації з форми, і metod, що приймає значення get або post. Ці значення

визначають спосіб передачі даних форми вузлу.

<INPUT> - це дескриптор відповідає за створення елементів управління,

розміщених у формі. Це дескриптор одиночного типу. З цим дескриптором

можуть використовуватись наступні атрибути:

----- type =“x” – визначає елемент управління;

----- name =”x” – ім’я елемента управління;

----- value =”x” – значення елемента управління; його атрибут є необов’язковим

для всіх елементів управління, крім перемикача (radio);

----- size =”x” – ширина поля елемента управління в пікселях; для елементів text і password ширину поля визначають в символах;

----- maxlength =”x” – максимальна кількість символів, котрі приймає елемент

управління;

----- checked =”x” – відмічений прапорець (перемикач); ----- src =”x” – вказівник на рисунок, що використовується в формі в якості

графічної кнопки.

<TEXTAREA>...</TEXTAREA> - цей елемент створює текстову область з

розширеними можливостями. Він схожий до елементів управління, але, крім

атрибутів, ми можемо управляти розміщений між дескрипторами текст. Крім

вже відомого атрибута name приймає атрибути rows=”x”, де “х” визначає

кількість стрічок області, і cols=”x”, де “x” визначає її ширину.

Верстка веб-сторінок - це верстка сайту на основі мови гіпертекстової розмітки (найчастіше використовується мова HTML), використанні каскадної таблиці стилів (CSS) та мови програмування JavaScript. Залежно від основних елементів html, що використовуються при верстці сайту, виділяють табличну верстку й блокову верстку. HTML (HyperText Markup Language) - мова гіпертекстової розмітки; стандартна мова розмітки веб-сторінок. Використовується для того, щоб додати необхідні елементи (зображення, текст та інше) на сторінку.

CSS (Cascading Style Sheets) - каскадна таблиця стилів; формальна мова опису зовнішнього вигляду веб-сторінки, яка містить розмітку. JavaScript – мова програмування, яка підтримує об’єктно-орієнтоване та функціональне програмування. Є реалізацією мови ECMAScript. Найчастіше використовується для створення інтерактивних веб-сторінок та виконується в браузері, а в оточенні Node.js ця мова використовується для створення серверної частини веб-застосунків.

Таблична верстка - це верстка сайту, в основі якої є таблиці і елементи сторінки розміщуються в комірках таблиці. Недоліки: відсутність кешування браузером, неможливо створити сучасний дизайн та важко адаптувати під різні розміри екранів, не відповідає правилам семантичної верстки.

Блокова верстка - це верстка сайту, в основі якої є блоки і за допомогою CSS елементи можна розмістити в будь-якому місці. Переваги: кешування браузером, можна створювати сучасний та складний дизайн, простота адаптування під різні пристрої.

Семантична верстка - це підхід до створення веб-сторінок на мові HTML, в основі якого є використання HTML-тегів у відповідності до їх призначення та дотримання логічної послідовності ієрархії сторінок. Наприклад: для кнопок використовувати <button> ...

</button> або <input type=“button”>, а тег <a href=“#”> ... </a> використовувати лише для посилань на інші веб-сторінки.

Таблиці стилів можуть створюватися і зберігатися окремо від тексту сторінки (у окремому файлі). Таким чином, операція форматування відділяється від змісту документа. У цьому і полягає основна ідея використання таблиць стилів (зміна зовнішнього вигляду без зміни змісту), завдяки чому зміст документа стає незалежним від пристроїв відображення. Передбачено декілька способів застосування стилів: таблиця стилів може бути поміщена в окремий файл, який зв'язується з HTML-документом (тег LINK ); таблицю стилів можна вбудувати в HTML-документ (тег STYLE ); стиль можна імпортувати з файлу або іншого документа; стиль можна задати для окремого елементу, якщо помістити у відповідний тег атрибут STYLE . Зв'язування документа з таблицею стилів, розміщеною у файлі. Розміщення таблиці в окремому файлі дозволяє використовувати її для декількох сторінок одночасно, забезпечити єдність стильового оформлення. Таблиця розміщується у файлі з розширенням .css. Файл текстовий, його можна створити за допомогою будь-якого текстового редактора. Для встановлення зв'язку з таблицею використовується тег LINK, наприклад: . Значення перших двох атрибутів цього тега є зарезервованими іменами, потрібними для того, щоб повідомити браузеру , що на цій сторічці використовуватиметься CSS . TYPE визначає синтаксис таблиць (text/css, більшість браузерів підтримують тільки цей, хоча може бути і інший, наприклад text/javascript) і дозволить браузерам, що не підтримують CSS, ігнорувати це посилання. Третій атрибут – HREF указує на файл, який містить описи стилів. Цей параметр повинен містити або відносний шлях до файлу (у випадку, якщо він знаходиться на тому ж сервері, що і документ, з якого до нього звертаються) або повний (у випадку, якщо файл стилів знаходиться на іншому сервері). Використання окремого файлу дозволяє створювати цілі сайти з однаковим стильовим оформленням. Вбудовування таблиці стилів в документ. Для включення таблиці стилів в документ використовується тег опис стилів Тег може містити атрибут TYPE, що набуває два значення (text/css або text/javascript). Вони забезпечують використання різних правил запису (різний синтаксис) таблиць. Розміщується  Відмінність імпорту від зв'язування полягає в тому, що при зв'язуванні таблиця використовується як альтернативна, а при імпорті вона додається до документа і має більший пріоритет у використанні. Завдання стиля для окремого фрагмента документа. Використовується для оперативного перевизначення стилю, визначеного в таблицях. Для цього у відповідному тегу задається атрибут STYLE. Наприклад, STILE=”color:blue”. Параметри стиля, задані в атрибуті, перекривають визначені поза тегом. Для визначення стиля частини тексту можна використовувати тег , для розділу документа –  (до речі, саме для цієї мети вони і вводилися).

**Верстка   веб-сторінок   - це** верстка   сайту   на   основі   мови гіпертекстової розмітки (найчастіше використовується мова HTML), використанні каскадної таблиці стилів (CSS) та мови програмування JavaScript. Залежно від основних елементів html, що використовуються при верстці сайту, виділяють табличну верстку й блокову верстку.

**HTML (HyperText Markup Language)** - мова гіпертекстової розмітки; стандартна мова розмітки веб-сторінок. Використовується для того, щоб додати необхідні елементи (зображення, текст та інше) на сторінку.

**CSS (Cascading Style Sheets)** - каскадна таблиця стилів; формальна мова опису зовнішнього вигляду веб-сторінки, яка містить розмітку.

**JavaScript** - мова програмування, яка підтримує об’єктно-орієнтоване та функціональне програмування. Є реалізацією мови ECMAScript. Найчастіше використовується для створення інтерактивних веб-сторінок та виконується в браузері, а в оточенні Node.js ця мова використовується для створення серверної частини веб-застосунків.

**Таблична верстка** - це верстка сайту, в основі якої є таблиці і елементи сторінки розміщуються в комірках таблиці. Недоліки: відсутність кешування браузером, неможливо створити сучасний дизайн та важко адаптувати під різні розміри екранів, не відповідає правилам семантичної верстки.

**Блокова верстка** - це верстка сайту, в основі якої є блоки і за допомогою CSS елементи можна розмістити в будь-якому місці. Переваги: кешування браузером, можна створювати сучасний та складний дизайн, простота адаптування під різні пристрої.

**Семантична верстка** - це підхід до створення веб-сторінок на мові HTML, в основі якого є використання HTML-тегів у відповідності до їх призначення та дотримання логічної послідовності ієрархії сторінок. Наприклад: для кнопок використовувати <button> … </button> або <input type=“button”>,  а  тег  <a  href=“#”>  …  </a>  використовувати  лише  для посилань на інші веб-сторінки.

**Верстка Mobile first** - це створення веб-сторінок спочатку для смартфонів, а потім адаптування їх для інших пристроїв. Тобто, верстка спочатку створюється для смартфонів, а вже потім для планшетів, ноутбуків та настільких пристроїв.

**Динамічні веб-сторінки** - це веб-сторінки, з якими можна

взаємодіяти та які містять динамічні елементи та динамічну інформацію.

Тобто, такі сторінки змінюють свій вигляд після взаємодії з ними

користувачів. Наприклад: список товарів в інтернет-магазині є

динамічним, оскільки товари можна додати, редагувати та видалити. А

інформація про інтернет магазин може бути статичною, тобто ніяк не

змінюватись.

**DOM** (Document Object Model - об’єктна модель документу) - це

об’єктна модель документу, яка представляє весь вміст сторінки у вигляді

об’єктів, які можна змінювати. Це всі елементи, які ми додаємо на

сторінку за допомогою мов розмітки: текст, зображення, відеозаписи,

форми, блоки та інше.