Лабораторна робота №4  
JavaScript та CSS

*Мета роботи:* розібрати способи застосування CSS на сторінках та динамічної зміни стилів за допомогою JavaScript.

*Знати:* особливості застосування CSS на сторінках та динамічної зміни стилів за допомогою JavaScript.

*Уміти:* підключати CSS; створювати CSS правила; працювати з CSS стилями через JavaScript сценарії; створювати анімацію використовуючи JavaScript та CSS.

*Обладнання:* технічне забезпечення: персональний комп’ютер, програмне забезпечення: текстовий редактор Notepad++, Visual Studio Code, браузер.

***Самостійна підготовка до виконання лабораторної роботи***

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту.
2. За літературою до лабораторної роботи вивчити необхідний теоретичний матеріал.
3. За інструкцією до лабораторної роботи ознайомитись з порядком виконання роботи.

***Короткі теоретичні відомості***

Cascading Style Sheet (CSS) – каскадні таблиці стилів, це технологія, яка використовується для стильового оформлення HTML-документа.

Каскадні таблиці стилів являють собою набір параметрів, які визначають положення та візуальне оформлення елементів в HTML-документі.

Переваги CSS:

* розмежування коду і оформлення;
* різне оформлення для різних пристроїв;
* розширені в порівнянні з HTML способи форматування;
* прискорення завантаження сайту;
* єдине стильове оформлення безлічі документів.

Способи підключення стилів:

1. Пов'язаний стиль – в окремому CSS-файлі, які пов'язаний зі сторінкою через елемент link, який доданий в head документа.
2. Глобальний стиль – створення стилю на сторінці в елементі style. Менш гнучкий спосіб ніж попередній, так як CSS правила доступні тільки одній сторінці.
3. Внутрішній стиль (inline) – стиль, який визначений в атрибуті style конкретного елемента розмітки.

Базовий синтаксис:

селектор { властивість1: значення; властивість2: значення; }

Приклади селекторів:

p {color: green} – зміна кольору тексту всіх параграфів;

.ok {color: green} – зміна кольору тексту елементів з атрибутом  
class = "ok";

#first {color: green} – зміна кольору тексту елементів з атрибутом id = "first";

p.ok {color: green} – зміна кольору тексту всіх параграфів з атрибутом class = "ok".

Dynamic HTML (DHTML) – це спосіб створення інтерактивного веб-додатку за допомогою комбінування статичної HTML розмітки, клієнтського JavaScript, CSS стилів та маніпуляцій з DOM.

Для того що б через JavaScript змінити оформлення елемента слід використовувати наступний код:

[Елемент].[Style].[Імя\_css\_атрібута] = "значення"

Наприклад:

e.style.color = "green";

e.style.width = "100px";

При використанні CSS атрибутів з ім'ям, яке містить "-" слід керуватися правилом – символ дефіса в назві слід видалити, а наступну літеру в назві CSS атрибута, яка йде після дефіса написати в верхньому регістрі (такий стиль іменування називається Camel Casing) див табл. 1.

*Таблиця 1*

|  |  |
| --- | --- |
| **CSS** | **JavaScript** |
| font-family | fontFamily |
| font-weight | fontWeight |
| font-style | fontStyle |
| background-color | backgroundColor |

***Порядок виконання роботи***

1. Вивчіть основні конструкції та поняття, які розглянуті на лекції.
2. Створіть сторінку. Додайте на сторінку 5 елементів div і кнопку. Створіть CSS файл, визначте в файлі правила, які зададуть усім тегам div наступне оформлення – висота і ширина 100 px, float = left та зелений колір фону.

Створіть сценарій, який буде обробляти натискання на кнопку і змінювати колір одного тега div на червоний. Зміна кольору повинна відбуватися циклічно, при першому натисканні перший тег div ставати червоний інші залишаються зеленими, при другому натисканні – всі теги div стають зеленими, а другий червоний і далі аналогічно.

1. Розробіть гру «Вгадай значення». Веб-сторінка загадує число від 1 до 100. Користувач в поле введення вводить значення намагаючись вгадати загадане число. Якщо користувач не вгадує значення сторінка виводить повідомлення з текстом «загадане значення більше введеного вами» або «загадане значення менше введеного вами». Подумайте як можна застосувати DHTML у такому додатку.
2. Додати сторінку для авторизації. На сторінці повинні знаходитися поля введення для логіна і пароля, кнопка «Вхід» та елемент checkbox «Запам'ятати мене». За допомогою CSS, який розташований в окремому файлі, вирівняйте форму для авторизації по центру сторінки. Розробіть сценарій, який буде спрацьовувати після натискання на кнопку «Вхід». Сценарій повинен реалізовувати такі дії:

* Якщо при натисканні на кнопку поля введення порожні – виводитися повідомлення (в тілі сторінки, не через alert) «Ви не заповнили поля: логін та пароль», також поля введення повинні отримати червоний фон.
* Якщо введений логін admin і пароль 12345 то відобразити користувачу зеленим кольором повідомлення «Ви авторизовані».

1. Розробіть сценарій, який щосекунди буде випадковим чином змінювати положення трьох елементів div. Задайте елементам фіксовану ширину і висоту та запустіть сценарій при завантаженні сторінки.
2. Розробіть сторінку, розмістить на ній чотири кнопки – «вгору», «вниз», «вліво» та «вправо». Помістіть на сторінці div з текстом. При натисканні на кнопки повинна запускатися анімація, яка переміщує div на 100 px у відповідному напрямку.

***Питання для самостійної роботи***

1. Опишіть призначення CSS?
2. Назвіть способи використання CSS на сторінці.
3. Які переваги дає використання CSS?
4. Опишіть призначення DHTML?
5. Які властивості CSS часто використовуються в DHTML?
6. Опишіть процес зміни фону елемента div при натисканні на кнопку.

***Література:***

1. Довідник по CSS: <http://htmlbook.ru/css>.
2. CSS для JavaScript-розробника: <http://learn.javascript.ru/object>.
3. Java Script and CSS: <https://developer.mozilla.org/ru/docs/Web/Guide/CSS/Getting_started/JavaScript>

Звіт до роботи

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, знати, уміти, обладнання та відповісти на запитання для самостійної підготовки.

2. Виконати завдання згідно свого варіанту.

3. Зробити висновок.