**Лабораторна робота №1**

**Тема. Вирішення логічних задач за допомогою мови програмування JavaScript. Застосування циклів, логічних галужень та робота з DOM.**

**Мета:** отримати практичні навички алгоритмічного розв’язання задач мовою JavaScript та роботи з DOM.

**Короткі теоретичні відомості**

1. **Що таке JavaScript?**

[**JavaScript**](https://developer.mozilla.org/uk/docs/Glossary/JavaScript) ("JS" скорочено) це повноцінна [динамічна мова програмування](https://developer.mozilla.org/en-US/docs/Glossary/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка, у застосуванні до [HTML](https://developer.mozilla.org/uk/docs/Glossary/HTML) документу, може надати динамічну інтерактивність на веб-сайтах. Вона була винайдена Бренданом Eйхом, співзасновником проекту Mozilla, the Mozilla Foundation, та Mozilla Corporation.

JavaScript має надзвичайно багато застосувань. Ви можете розпочати з малого: створити "каруселі", галереї зображень, динамічні макети сторінок, відповіді на натиски кнопок, тощо. Із досвідом, ви зможете створювати ігри, 2D та 3D графіку, складні застосунки з використанням баз даних та багато іншого.

JavaScript доволі компактна та гнучка мова. Розробники забезпечили велике розмаїття інструментів, що доповнюють основу мови JavaScript, які відкривають величезну кількість додаткового функціоналу з мінімальними зусиллями. Серед них:

* Програмні інтерфейси ([APIs](https://developer.mozilla.org/uk/docs/Glossary/API)) для браузерів — API, які вбудовані у браузери, що надають функціонал на зразок динамічного створення HTML та застосування CSS-стилів, збір та обробка відео-потоків з вебкамери користувача, генерація 3D-графіки та аудіо-семплів.
* API третіх осіб, що дозволяють розробникам інтегрувати у власні сайти функціонал інших провайдерів, таких як Twitter або Facebook.
* Фреймворки та бібліотеки третіх осіб, які ви можете застосувати до вашого HTML, щоб прискорити створення сайтів та застосунків.

1. **Оголошення змінних.**

Для зберігання даних у програмі використовуються змінні. Змінні призначені для зберігання яких-небудь тимчасових даних або таких даних, які в процесі роботи можуть змінити своє значення. Для створення змінних застосовується ключове слово var:

var myIncome;

За допомогою знаку дорівнює можна присвоїти змінній будь-яке значення:

var income = 300;

1. **Умовні конструкції**

**Вираз if**

Констукція if перевіряє деяку умову та якщо ця умова вірна, то виконує деякі дії:

var income = 100;

if(income > 50){

var message = "доход больше 50";

alert(message);

}

### Конструкція switch..case

Конструкція switch..case є альтернативою конструкції if..else if..else та також дозволяє обробити декілька умов:

var income = 300;

switch(income){

    case 100 :

        console.log("Доход равен 100");

        break;

    case 200 :

        console.log("Доход равен 200");

        break;

    case 300 :

        console.log("Доход равен 300");

        break;

}

1. **Робота з DOM**

Для пошуку елементів на сторінці застосовуються наступні методи:

* **getElementById(value):** вибирає елемент, у якого атрибут id дорівнює value
* **getElementsByTagName(value):** всі елементи, у яких тег дорівнює value
* **getElementsByClassName(value):** всі элементи, які мають клас value
* **querySelector(value):** перший елемент, який відповідає css-селектору value
* **querySelectorAll(value):** всі елементи, які відповідають css-селектору value

1. **Події**

Для взаємодії з користувачем у JavaScript визначено механізм подій. Наприклад, коли користувач натискає кнопку, то виникає подія натискання кнопки. У коді JavaScript ми можемо визначити виникнення події та якось обробити їі.

Приклад створення події натискання на елемент:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

</head>

<body>

<div id="rect" onclick="displayMessage()"></div>

<script>

function displayMessage(){

    alert('Нажато');

}

</script>

</body>

</html>

При натисканні на елемент div спрацює функція displayMessage(), яка виведе повідомлення на екран.

**Завдання до самостійної роботи**

У даній роботі необхідно дотримуватися наступного алгоритму:

1. Створіть веб-сторінку мовою HTML (index.html) у будь-якому текстовому редакторі та додайте три текстові поля (<input type="text" maxlength="25" size="20" id="myVar"> ) для введення чисел згідно із завданням та кнопку  “Обчислити” (<input type="button" value="Обчислити"/>).  Назва текстового поля (id) має бути унікальним. Для отримання доступу до значення текстового поля в Javascript використовуйте конструкцію document.getElementById(“myVar”).value.
2. Створіть файл Javascript (myscript.js) у будь-якому текстовому редакторі.
3. Підключіть файл myscript.js до вашої веб-сторінки (index.html) за допомогою конструкції <script type="text/javascript" src="myscript.js"></script> . Цю конструкцію (script) треба додати до розділу head веб-сторінки.
4. Реалізуйте додаток мовою Javascript згідно з варіантом вашого завдання та додайте його до файла myscript.js. Для цього створіть функцію myFunction у файлі myscript.js та підключіть її до обробника натискання (onclick) кнопки «Обчислити», наприклад, за допомогою конструкції <input type="button" onclick="myFunction()" value="Обчислити"/>  (файл index.xhtml).
5. Запустіть на виконня скрипт  myscript.js. Для цього треба веб-сторінку index.html завантажити до веб-браузера, наприклад, до Firefox.
6. Перевірте роботоспроможність додатка згідно з вимогами вашого завдання та у випадку необхідності внесіть відповідні коригування коду скрипту вашого додатка.
7. Підготуйте звіт про виконану лабораторну роботу.

**Варіанти завдань**

1. Розробіть додаток, який дозволяє обчислити суму трьох чисел у випадку, коли доданки більші за 1. Додаток  повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та суму виведіть на екран.
2. Розробіть додаток, який дозволяє обчислити різницю трьох чисел у випадку коли перше число більше другого на 2. Додаток  повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та різницю виведіть на екран.
3. Розробіть додаток, який дозволяє поділити три числа у випадку коли друге і третє число більше за 0. Додаток  повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат вводу та ділення виведіть на екран.
4. Розробіть додаток, який дозволяє перемножити три числа у випадку коли кожне із чисел менше за -1. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та множення виведіть на екран.
5. Розробіть додаток, який дозволяє знайти суму квадратів трьох чисел у випадку, коли перше і третє число більше за 0. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та максимальне число виведіть на екран.
6. Розробіть додаток, який дозволяє знайти суму кубів чисел у випадку коли перше число дорівнює другому. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та множення виведіть на екран.
7. Розробіть додаток, який дозволяє замінити всі введені 0 на 1, а всі від'ємні на їх модуль. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури. Результат введення та заміни виведіть на екран.
8. Розробіть додаток, який дозволяє збільшити на 1 кожне з уведених чисел, якщо перше і друге число від'ємні. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та заміни виведіть на екран.
9. Розробіть додаток, який дозволяє перемножити тільки три числа з плаваючою точкою і які більші за 1.32. Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат введення та перемноження виведіть на екран.
10. Розробіть додаток, який дозволяє змінити місцями значення  першого — третього полів таким чином, щоб перше число було більше за друге, а друге —більше за третє.  Додаток повинен бути інтерактивним, тобто числа треба ввести за допомогою клавіатури.  Результат вводу та заміни виведіть на екран.

**Контрольні питання**

* + 1. Що таке Javascript? Дайте стислу характеристику цієї мови.
    2. Яке застосування мови Javascript?
    3. Синтаксис підключення скриптів JavaScript до веб-сторінки.
    4. Умовні конструкції мови Javascript.
    5. Синтаксис написання функцій мовою Javascript.
    6. Які математичні операції підтримуються мовою Javascript?

**Лабораторна робота №2**

**Тема. Об’єкт Node. Навігація по DOM.**

**Мета:** отримати практичні навички з об’єктами Node та навігацією по DOM.

**Короткі теоретичні відомості**

Кожен окремий вузол, будь то html-елемент, його атрибут або текст, в структурі DOM представлений об'єктом Node. Цей об'єкт надає ряд властивостей, за допомогою яких ми можемо отримати інформацію про даний вузол:

* **childNodes**: містить колекцію дочірніх вузлів
* **firstChild**: повертає перший дочірній вузел поточного вузла
* **lastChild**: повертає останній дочірній вузел поточного вузла
* **ownerDocument**: повертає корневий вузел документу
* **parentNode**: повертає елемент, що містить поточний вузел
* **nodeName**: повертає ім’я вузла
* **nodeType**: повертає тип вузла у вигляді числа
* **nodeValue**: повертає або встановлює значення вузла у вигляді простого тексту

Для створення елементів об’єкт document має наступні методи:

* **createElement(elementName):** створює елемент html, тег якого передається у якості параметра. Повертає створений елемент.
* **createTextNode(text):** створює та повертає текстовий вузел. В якості параметра передається текст вузла.

var elem = document.createElement("div");

var elemText = document.createTextNode("Привет мир");

Після створення об’єкта, його необхідно додати до існуючих елементів. Для додавання елементів ми можемо використовувати один із методів об’єкта Node:

* **appendChild(newNode):** додає новий вузел newNode в кінець колекції дочірніх вузлів
* **insertBefore(newNode, referenceNode):** додає новий вузел newNode перед вузлом referenceNode

**Завдання до самостійної роботи**

У даній роботі необхідно дотримуватися наступного алгоритму:

* 1. Розробити сторінку HTML за своїм варіантом та додати дві кнопки та поля вводу для кожного стовпчика таблиці. Перша кнопка­ - «Додати», при натисканні якої введені дані з полів додаються як рядок у кінець таблиці. Інша кнопка – «Видалити», при натисканні якої видалиться останній рядок таблиці.
  2. Для цього необхідно створити скрипт у файлі myscript.js з функцією додавання рядка до таблиці та функцією видалення.
  3. Для створення та видалення строк можна використовувати методи insertRow(), insertCell() та deleteRow().
  4. Підготуйте звіт про виконану лабораторну роботу.

**Варіанти завдань**

1. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка повинна містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства.  На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля комп’ютерами.
2. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля книгами.
3. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля журналами.
4. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля побутовою технікою.
5. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля мобільними пристроями.
6. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля прикрасами.
7. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства.  На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля торговельним обладнанням.
8. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля продовольчими товарами.
9. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства. На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля одягом.
10. Розробіть дві веб-сторінки мовою HTML торговельного підприємства та автора. Перша веб-сторінка має містити інформацію про підприємство (назву, адресу, телефони офісу, час роботи) та прайс-листок підприємства.  На другій веб-сторінці подайте відомості про розробника. Діяльність підприємства – торгівля радіоприладами.

**Контрольні питання**

1. Що таке об’єкт Node? Його властивості.
2. Які методи document існують для створення елементів?
3. Які методи існують для додавання створеного елемента до колекції?

**Лабораторна робота №3**

**Тема. Робота з JSON.**

**Мета:** отримати практичні навички роботи з JSON форматом.

**Короткі теоретичні відомості**

JSON (JavaScript Object Notation) представляє легкий формат зберігання даних. JSON описує структуру і організацію даних JavaScript. Простота JSON привела до того, що в даний момент він є найбільш популярним форматом передачі даних в середовищі web, витіснивши інший колись популярний формат xml. Об'єкти JSON дуже схожі на об'єкти JavaScript, тим більше що JSON є підмножиною JavaScript. У той же час важливо їх розрізняти: JavaScript є мовою програмування, а JSON є форматом даних.

JSON підтримує три типи даних: примітивні значення, об'єкти і масиви. Примітивні значення представляють стандартні рядки, числа, значення null, логічні значення true і false.

Об'єкти представляють набір найпростіших даних, інших об'єктів та масивів. Наприклад, типовий об'єкт JSON:

{

    "name": "Tom",

    "married": true,

    "age": 30

}

У javascript цьому об'єкту відповідав би наступний:

var user = {

    name: "Tom",

    married: true,

    age: 30

}

Незважаючи на загальну схожість, в той же час є і відмінності: в JSON назви властивостей розміщують в лапки, як звичайні строки. Крім того, об'єкти JSON не можуть зберігати функції, змінні, як об'єкти javascript.

### Серіалізація в JSON та десеріалізація

Для серіалізації об’єкта у javascript в json застосовується функція JSON.stringify():

var user = {

    name: "Tom",

    married: true,

    age: 30

};

var serializedUser = JSON.stringify(user);

document.write(serializedUser); // {"name":"Tom","married":true,"age":30}

Для зворотної операції - десеріалізації або парсинга json-об'єкта в javascript застосовується метод JSON.parse ():

var user = {

    name: "Tom",

    married: true,

    age: 30

};

// серіалізація

var serializedUser = JSON.stringify(user);

// десеріалізація

var tomUser = JSON.parse(serializedUser);

document.write(tomUser.name); // Tom

**Доступ до JSON файлу за URL використовуючи AJAX**

Створюємо об’єкт XMLHttpRequest():

let request = new XMLHttpRequest();

Відкриваємо файл використовуючи GET запит:

request.open('GET', 'users.json', true);

Замість параметру 'users.json' можете використовувати будь-яку URL адресу.

Парсимо наші JSON дані у функції onload. Після цього перебираємо дані в циклі та відобразимо користувачу в елементі ul.

request.onload = function () {   
let users = JSON.parse(this.response);

let output = '';  
 for (*var* i = 0; i < users.length; i++) {  
 output += '<li>' + users[i].name + ' is ' + users[i].age + ' years old.'; '</li>'  
 }  
 document.getElementById('users').innerHTML = output;  
}

Відправляємо запит:

request.send();

Кінцевий код для прикладу:

<body>

<div id="users"></div>

<script>

let request = new XMLHttpRequest();

request.open('GET', 'http://jsonplaceholder.typicode.com/posts', true);

request.onload = function () {

let users = JSON.parse(this.response);

let output = ' List of users';

for (var i = 0; i < users.length; i++) {

output += '<li>' + users[i].userId + ' | ' + users[i].title + ' |'; '</li>'

}

document.getElementById('users').innerHTML = output;

}

request.send();

</script>

**Завдання до самостійної роботи:**

Розробити HTML сторінку за варіантом завдання, поданого у лабораторній роботі №2, завантаживши дані прайс-листа з JSON формату.

**Контрольні питання**

* 1. Що таке JSON? Його особливості.
  2. Чим відрізняється об’єкт JSON від об’єкту JavaScript?
  3. Які методи використовуються для сериалізації та дисеріалізації JSON?
  4. Яка технологія дозволяє отримати JSON файл використовуючи GET запит?

**Лабораторна робота №4**

**Тема. Зберігання даних у JavaScript. Робота з Local storage.**

**Мета:** отримати практичні навички роботи з local storage.

**Короткі теоретичні відомості**

**Зберігання даних. Куки**

Одну з можливостей збереження даних в javascript представляє використання куки. Для роботи з куками в об'єкті document призначена властивість cookie.

Для установки куків досить властивості document.cookie привласнити рядок з куками:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

</head>

<body>

<script>

document.cookie = "login=tom32;";

</script>

</body>

</html>

Рядок куки приймає до шести різних параметрів: ім'я куки, значення, термін закінчення дії (expires), шлях (path), домен (domain) і secure. Вище використовувалися тільки два параметри: ім'я куки і значення. Тобто в разі з рядком "login = tom32;" куки має ім'я login і значення tom32.

Але подібна куки має дуже обмежений термін життя: якщо явно не встановити термін дії, то кука буде вилучена із закриттям браузера. Подібна ситуація, можливо, ідеальна для тих випадків, коли необхідно видаляти всю інформацію після завершення роботи з веб-додатком і закриття браузера. Однак дане поведінка не завжди підходить.

І в цьому випадку нам треба встановити параметр expires, тобто термін дії куків:

document.cookie = "login=tom32;expires=Mon, 31 Aug 2015 00:00:00 GMT;";

Якщо в нам треба встановити куки для якогось певного шляху на сайті, то ми можемо використовувати параметр path. Якщо на нашому сайті є кілька доменів, і ми хочемо встановити куки безпосередньо для певного домену, тоді можна використовувати параметр domain.

Останній параметр - secure задає використання SSL (SecureSockets Layer) і підходить для сайтів, які використовують протокол https. Якщо значення цього параметра дорівнює true, то куки будуть використовуватися тільки при установці захищеного з'єднання ssl. За замовчуванням цей параметр дорівнює false.

**Web storage**

**Web storage** складається з двох компонентів: **session storage** і **local storage**.

**Session storage** представляє тимчасове сховище інформації, яка видаляється після закриття браузера.

**Local storage** представляє сховище для даних на постійній основі. Дані з local storage автоматично з сайту не видаляються і не мають терміну дії. Ці дані не передаються на сервер в запиті HTTP. Крім того, обсяг local storage становить в Chrome і Firefox 5 Mб для домену, а в IE - 10 Mб. Всі дані в web storage представляють набір пар ключ-значення. Тобто кожен об'єкт має унікальне ім'я-ключ і певне значення.

Для роботи з local storage в javascript використовується об'єкт localStorage, а для роботи з session storage - об'єкт sessionStorage. Для збереження даних необхідно передати в метод setItem () об'єкта localStorage:

localStorage.setItem("login", "admin");

У цей метод передаються два значення: ключ і значення об'єкту, що зберігається. Якщо в localStorage вже є об'єкт з ключем "login", то його значення замінюється новим.

Для отримання збережених даних треба викликати метод getItem ():

var login = localStorage.getItem("login");

Щоб видалити об'єкт, застосовується метод removeItem (), який приймає ключ об'єкта, що видаляється:

localStorage.removeItem("login");  
І для повного видалення всіх об'єктів з localStorage можна використовувати метод clear ():

localStorage.clear();

**Завдання до самостійної роботи:**

Створити toDo ліст на JavaScript із збереженням даних у local storage. Додаток  повинен бути інтерактивним, тобто дані треба ввести за допомогою клавіатури.

**Контрольні питання**

* 1. Які способи зберігання даних існують у JavaScript?
  2. Для чого використовується web storage?
  3. Яка різниця між session та local storage?
  4. Які методи застосовуються для збереження та доступу до даних у local storage?
  5. Як повністю видалити всі об’єкти, що знаходяться у local storage?

**Лабораторна робота №5**

**Тема. Ефекти та анімація у jQuery.**

**Мета:** отримати практичні навички роботи з ефектами jQuery.

**Короткі теоретичні відомості**

**Початок роботи з jQuery**

jQuery це швидка, невелика і багата можливостями JavaScript бібліотека. Вона дозволяє дуже просто робити такі речі як: обхід елементів або маніпуляція елементами HTML документа або AJAX запити. Не хвилюйтесь про крос-браузерність - це і багато іншого бере на себе jQuery.

**Підключення бібліотеки jQuery**

Щоб почати працювати з цією бібліотекою, нам насамперед треба її завантажити. Її знайти можна на офіційному сайті розробника <https://jquery.com/download/>.

Бібліотека представлена ​​в двох варіантах - Compressed або Monified (мінімізована) і Uncompressed (звичайний). Мінімізовані версії надають ту ж функціональність, що і звичайні, але відрізняються тим, що не містять будь-яких необов'язкових символів, на зразок прогалин, коментарів і т.д., і тому в своїй назві маю суфікс min, наприклад, jquery-1.10.1.min .js. Оскільки вони продуктивніше за рахунок меншого обсягу, їх рекомендується використовувати в реальному виробництві. У той же час, якщо вам захочеться зрозуміти логіку коду jQuery, то в цьому випадку можна звертатися до звичайної версії бібліотеки.

Бібліотека jquery підключається також, як і інші файли javascript. Наприклад:

<!DOCTYPE html>

<html>

<head>

<title>Мир jQuery</title>

<script src="jquery-1.10.1.min.js"></script>

</head>

<body>

</body>

</html>

У попередньому прикладі бібліотеку jquery підключено безпосередньо з локального диска, проте нам необов'язково завантажувати бібліотеку і розміщувати на локальному диску поруч з іншими файлами. Замість цього ми можемо використовувати мережі CDN (Content Delivery Networks). В даному випадку сама бібліотека буде знаходиться фізично в будь-якої мережі CDN, а ми можемо вказати на неї посилання.

Наприклад, підключимо бібліотеку jquery, що знаходиться в jQuery CDN:

<!DOCTYPE html>

<html>

<head>

<title>Мир jQuery</title>

<script src="<https://code.jquery.com/jquery-1.10.1.min.js>"></script>

</head>

Необов'язково підключати саме цю версію бібліотеки, весь набір доступних версій бібліотеки можна знайти за адресою <https://code.jquery.com/>.

**Базові ефекти**

До базових ефектів в jQuery відносяться ефекти приховування і відображення елементів, які досягаються за допомогою методів show(), hide() і toggle().

Так, наприклад, ми можемо приховувати і відображати елементи при натисканні на кнопку:

<ul>

    <li>Java</li>

    <li>C/C++</li>

    <li>JavaScript</li>

</ul>

<button id="show">Показать</button>

<button id="hide">Скрыть</button>

<script type="text/javascript">

$(function() {

    $('button#show').click(function(){

        $('ul').show();

    });

    $('button#hide').click(function(){

        $('ul').hide();

    });

});

</script>

Ефекти ковзання дозволяють нам плавно приховати або розкрити елемент. Ефекти ковзання реалізовані у вигляді методів slideUp(), slideDown() і slideToggle().

Ці методи мають однакові форми використання:

– slideUp/slideDown/slideToggle(): метод без параметрів;

* + slideUp/slideDown/slideToggle([duration] [, easing][, complete]). Параметр duration вказує як довго приховування елемента буде тривати. За замовчуванням його значення дорівнює 400 мілісекунд.

Ефекти прозорості дозволяють нам, плавно змінюючи прозорість елемента, приховати його або відобразити. Ефекти прозорості реалізовані за допомогою методів fadeOut(), fadeIn(), fadeTo() и fadeToggle().

|  |  |
| --- | --- |
| **Метод** | **Опис** |
| fadeOut | Приховує елемент, зменшуючи його прозорість |
| fadeIn | Відображає елемент, збільшуючи його прозорість |
| fadeToggle | Поєднує методи fadeOut и fadeIn: якщо прозорість дорівнює нулю, то елемент відображується. Якщо елемент непрозорий, то він приховується |
| fadeTo | Зміна прозорості до вказаного рівня |

Метод animate () приймає набір властивостей елемента, які потім змінюються, за рахунок чого досягається анімація. Даний метод має таку форму використання: animate (properties [, duration] [, easing] [, complete]).

Обов'язковий параметр properties містить набір css-властивостей, у яких вказуються фінальні значення. Параметр duration вказує, як довго буде тривати зміна прозорості елемента. За замовчуванням його значення дорівнює 400 мілісекунд. Також можна використовувати значення 'slow' і 'fast', які відповідають тривалості ефекту в 600 і 200 мілісекунд. Параметр easing приймає назву функції плавності анімації у вигляді рядка. За замовчуванням його значення дорівнює "swing". Параметр complete представляє функцію зворотного виклику, що викликається методом по завершенні анімації.

Використовуючи анімацію, важливо мати на увазі, що в даному випадку ми можемо використовувати тільки ті властивості css, які приймають числові значення, наприклад, ширина і висота. Інші ж властивості, як, наприклад, колір шрифту, ми вже так просто не зможемо використовувати.

**Завдання до самостійної роботи:**

До HTML сторінки, створеної у лабораторній роботі №2, додати декілька ефектів jQuery. Можна використовувати базові ефекти, описані у теоретичній частині або ефекти з бібліотеки jQuery UI.

**Контрольні питання**

1. Які можливості має бібліотека jQuery?
2. Які існують способи підключення бібліотеки jQuery?
3. Які базові ефекти існують в бібліотеці jQuery?
4. Які методи дозволяють створювати більш складні анімації?