Лабораторна робота №4

**Підключення користувацьких сценаріїв до вебсторінок**

**Мета** Навчитися підключати та використовувати сценарії JavaScript у вебдокументах.

Теоретичні відомості JavaScript (JS) — динамічна, об'єктно-орієнтованa мова програмування. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні серверу (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

При використанні в рамках технології DHTML JavaScript код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги з обов'язковим по специфікації HTML 4.01 атрибутом type="text/javascript", хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript. Javascript підключається безпосередньо в HTML-файл. Найпростіший спосіб — це написати javascript-команди всередину тега де-небудь в тілі сторінки. Підключення в будь-якому місці документу Коли браузер читає HTML-сторінку — він першим ділом читає і виконує код, а тільки потім продовжує читати сторінку далі. Так, у наступному прикладі буде показано початок сторінки, потім три рази виконається функція alert, яка виводить віконце з інформацією, а тільки потім з'явиться інша частина сторінки.

У цьому прикладі використовувалися наступні елементи. ... Тег повідомляє браузеру про те, що всередині знаходиться виконуваний скрипт. Атрибут type говорить про те, що це javascript. Взагалі кажучи, атрибут може бути відсутнім — різниці немає, але з точки зору стандарту його слід вказати. Конструкція for — Звичайний цикл, по синтаксису аналогічний іншим мовам програмування. Оголошення var — Оголошення змінної всередині циклу: i — локальна змінна. Функція alert — Виводить повідомлення на екран і чекає, поки відвідувач не натисне ОК Підключення в заголовку HEAD Зазвичай javascript намагаються відокремити від власне документа. Для цього його поміщають всередину тега HEAD, а в тілі сторінки по можливості залишається чиста верстка. У наступному прикладі javascript-код тільки описує функцію count\_rabbits, а її виклик здійснюється після натискання на кнопку input. function count\_rabbits () { for (var i = 1; i <= 3; i + +) { / / Оператор + з'єднує рядки alert ("З капелюха дістали" + i + "кролика!") } }

Для вказівки запускається по кліку функції в input був використаний атрибут onclick. Це лише вершина потужного айсбезга javascript-подій. Зовнішні сценарії Зазвичай javascript-код взагалі не пишуть в HTML , а підключають окремий файл зі скриптом: При цьому файл /my/script.js містить javascript -код , який інакше міг би перебувати всередині тега . Це дуже зручно, тому що один і той же файл з скриптами можна підключати на різних сторінках. При правильних налаштуваннях сервера браузер кешує його і не буде скачувати щоразу заново. Щоб підключити декілька скриптів — використовуйте кілька таких тегів: ... Тобто одночасно підключити зовнішній файл і написати щось всередині тега можна. Доведеться робити два різних тега: перший з src , другий — з командами, які будуть виконані після виконання зовнішнього файлу .

**Завдання**

Виконати підключення сценарію Javascript, який реалізовує певний функціонал на сайті. Описати функції, що виконуються сценарієм, спосіб підключення, навести результат виконання.

**Код програми**

<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8">

  <title>JS Bin</title>

</head>

<body>

  <input type="number" id="num">

  from <select id="from"></select> to

  <select id="to"></select> =

  <span id="result"></span>

  <script>

    var Money = function(symbol, value){

  this.symbol = symbol;

  this.value = value;

}

var arr = [

  new Money("$", 27.2),

  new Money("₽", 3.33333),

  new Money("¥", 0.24),

  new Money("₴", 1)

];

var numInput = document.getElementById("num");

var fromSelect = document.getElementById("from");

var toSelect = document.getElementById("to");

var resultSpan = document.getElementById("result");

arr.forEach(function(money){

  var option = document.createElement("option");

  option.text = money.symbol;

  fromSelect.add(option);

  option = option.cloneNode();

  option.text = money.symbol;

  toSelect.add(option);

});

numInput.oninput = fromSelect.oninput = toSelect.oninput = function(event){

  var money1 = arr[fromSelect.options.selectedIndex];

  var money2 = arr[toSelect.options.selectedIndex];

  var num = +numInput.value;

  if(!numInput.value.length || isNaN(num)){

    return;

  }

  console.log(numInput.value.length)

  var result = money1.value / money2.value \* num;

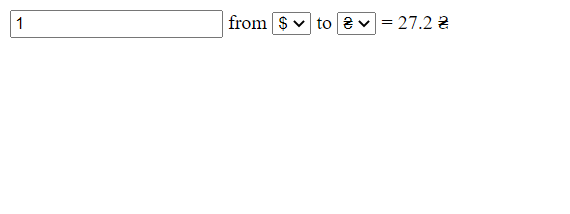
  resultSpan.innerHTML = Math.round(result \* 1000) / 1000 + " " + money2.symbol;

}

  </script>

</body>

</html>

Результат виконаго коду

Висновок: навчився підключати та використовувати сценарії JavaScript у вебдокументах.