

Лекція 27.



WEB. JavaScript. Призначення, основні поняття,
визначення.



План лекції

- Що таке скрипт?
- Що можуть скрипти.
- Приклади використання скриптів.
- Розміщення скриптів в HTML.
- Об'єктна модель документу (DOM).
 - представлення DOM.
 - представлення документу в DOM.
- Доступ до об'єктів.
 - імена об'єктів.
 - посилання на об'єкти.



Що таке скрипт?

- JavaScript — скриптова мова, яка найчастіше використовується при створенні сценаріїв поведінки браузера, що вбудовуються у веб-сторінки. Дозволяє динамічно змінювати вигляд веб-сторінки з боку клієнта (тобто, без пересилання інформації серверу).
- Скриптова мова (англ. scripting language, рос. язык сценариев) — мова програмування, що розроблена для запису «сценаріїв» - послідовності операцій, які користувач може виконувати на комп'ютері. Як правило, сценарії інтерпретуються, а не компілюються.
- Особливості скриптів:
 - програмування без ризику дестабілізувати систему. (+)
 - кросплатформеність (виконуються на різних браузерах) (+)
 - виконуються на льоту, без компіляції, що призводить до меншої швидкості виконання. (-)



Що можуть скрипти?

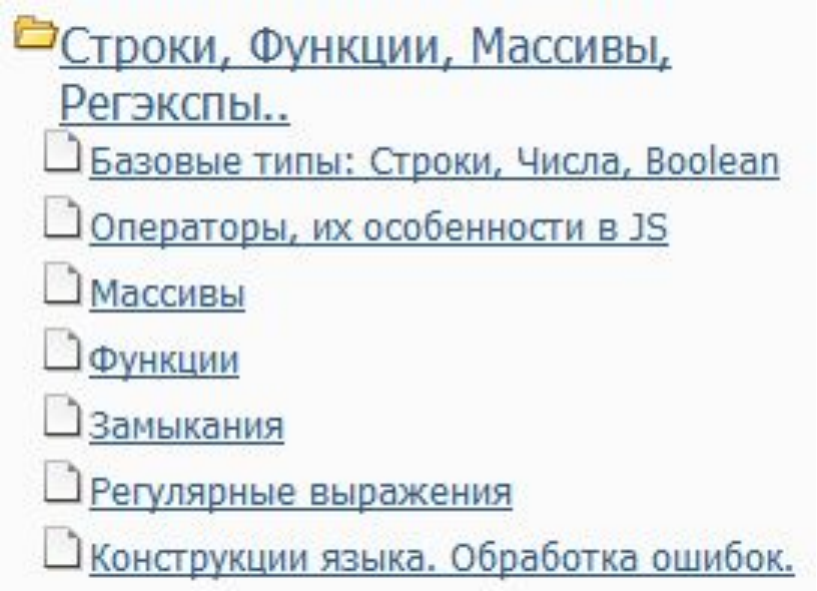
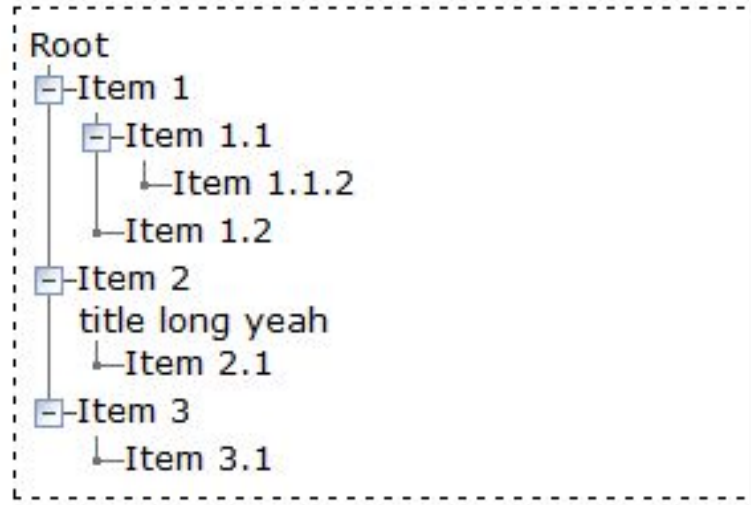
- Можуть виконуватись при завантаженні документа, й динамічно змінювати його зміст («Добрий ранок!», «Добрий день!»...).
- Можуть використовуватись для реакції на такі події, як відкриття чи закриття документа, рух миші, натискання клавіш на клавіатурі й т. п.
- Можуть використовуватись для обробки даних, які вводяться користувачем в елементи форми. (Заповнення полів на основі інших полів, контроль правильності введення даних).
- Можуть під'єднуватись до елементів форм (кнопки, списки, перемикачі) для створення графічного інтерфейсу користувача.
- Можуть використовуватись для створення інтерактивного інтерфейсу. (створення деревоподібного ієрархічного меню – «дерева перегляду»).

Приклади використання JavaScript

- Деревовидне меню

1)

2)



Приклади використання JavaScript

- Веб-калькулятор кольорового кодування резисторів.

Calculate Resistor Values from Color Codes

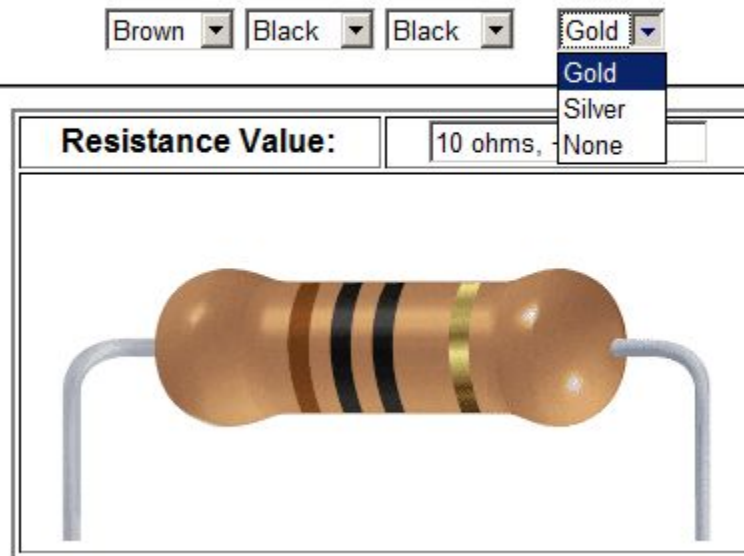
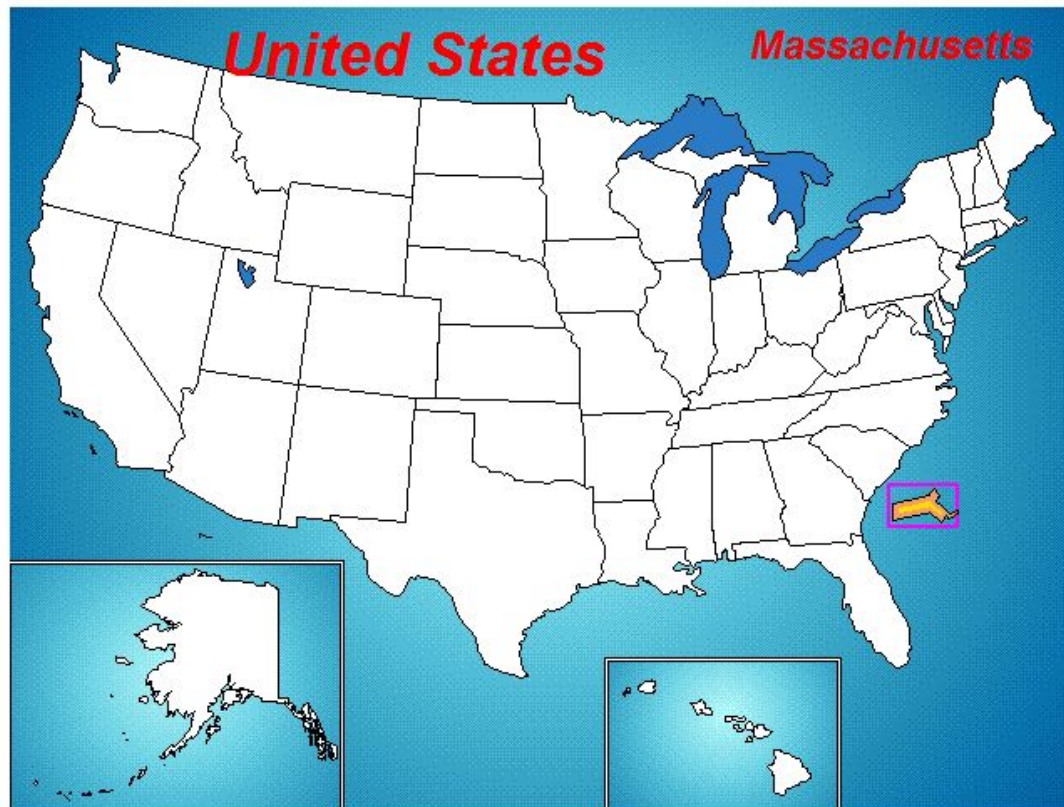


Illustration Copyright 1996 Danny Goodman (AE9F). All Rights Reserved.

Приклади використання JavaScript

- Навіть прості ігри...

United States Map Puzzle



Click and drag the puzzle piece to the appropriate place on the map.



Розміщення скриптів в HTML.

JavaScript-код вбудовується в HTML-код сторінки й виконується інтерпретатором, що вбудований в браузер.

- Вбудовані скрипти:

```
<script type="text/javascript">  
  alert('Hello, World!');  
</script>
```

- Інтегровані в атрибути подій:

```
<a href="delete.php" onclick="return confirm('Вы  
уверены?');">Удалить</a>
```

- Зовнішні скрипти (в окремому файлі, що підключається до HTML-файлу):

```
<script type="text/javascript" src="../script/load.js"></script>
```

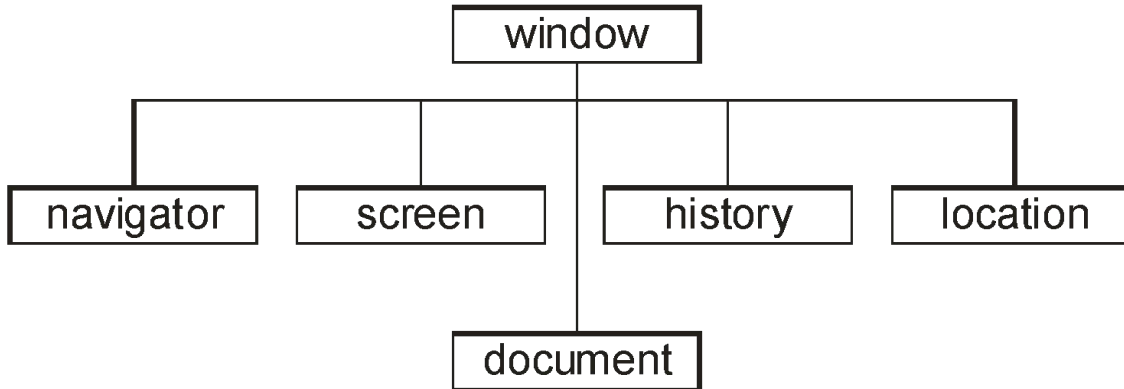



Об'єктна модель документа

- DOM (Document Object Model) – об'єктна модель документа – це не залежний від платформи і мови програмний інтерфейс, що дозволяє програмам і скриптам отримувати доступ до змісту документів, і змінювати структуру й оформлення документів.
 - DOM – структура організації об'єктів у вікні.
- Будь-який документ за допомогою DOM може бути представлений як дерево вузлів.
 - Кожен вузол – елемент, атрибут, текст, зображення й т.п.
 - Вузли пов'язані один з одним за допомогою співвідношень "родительський-дочерний".

Представлення DOM.

Загальна ієрархічна структура DOM:

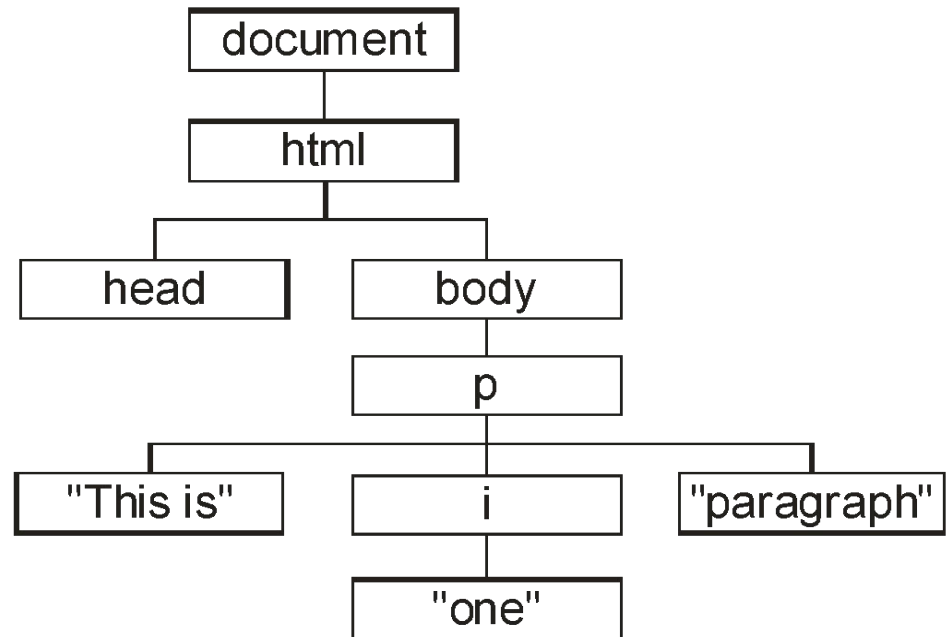


- Window – частина вікна браузера, де відображається документ.
- Navigator – визначає браузер, в якому виконується сценарій. Має атрибут “тільки читання”.
- Screen – визначає фізичне середовище, де запускається браузер (напр. роздільна здатність екрану).
- History – історія відкритих сторінок, стек.
- Location – дозволяє завантажити сторінку у вікно.
- Document – завантажений документ, містить всі інші об’єкти.

Представлення документа в DOM.

Представлення відкритого документа у вигляді об'єктів:

```
<html>  
  <head></head>  
  <body>  
    <p>This is  
    <i>one</i>  
    paragraph</p>  
  </body>  
</html>
```





Доступ до об'єктів.

Імена об'єктів.

- найчастіше – використовують ідентифікатори:
 - `<p id="text1">`
 - `<div class="all" id="key">`
 - ``
- особливості імен:
 - не можуть містити прогалики, символи пунктуації, не можуть починатись з цифри.

Посилання на об'єкт:

- `window.document.getElementById("elementID").власивість`
 - кожне посилання починається з найбільш глобального рівня
 - крапка – розділяє рівні ієрархії.
 - кожен об'єкт має свій набір вкладених об'єктів (їх досить багато).



Властивості і методи.

Властивості об'єктів – можна порівняти з прикметниками, для кожного об'єкта може бути декілька властивостей:

- `document.getElementById("table1").align`
- `document.getElementById("row").bgcolor`

Методи об'єктів – можна порівняти з дієсловами:

- `document.getElementById("form_key").submit()` – відправка даних з форми при клацанні ЛК миші на кнопці.
- `document.getElementById("entry").focus()` – виділення тексту в текстовому полі.

Методу може передаватись додаткова інформація – параметр або аргумент:

- `document.write(" of " + navigator.appName + ".")` – на екран виводиться ім'я браузера.



Обробка подій.

Обробка подій – реакція скрипта на певні дії, які робить користувач, або ж, які виконує сам браузер (завантаження сторінки, повідомлення про помилки і т.п.).

Як правило, такими, що найбільш широко використовуються, є події, які робить користувач:

- `<input type="button" value="click me" onclick="window.alert('натиснута кнопка!')">`