

Лекція 8. Основи JavaScript

Характеристика та властивості JavaScript

JavaScript — назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови — написання сценаріїв для веб-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

Коротка характеристика

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

Об'єкт називають «об'єктом першого класу» коли він:

1. може бути збережений у змінній
2. може бути переданий у функцію як параметр
3. може бути повернутий з функції як результат
4. може бути створений під час виконання програми внутрішньо самоідентифікуємий (незалежний від іменування)

Динамічна типізація — прийом, широко використовуваний у мовах програмування й мовах специфікації, при якому змінна зв'язується з типом у момент присвоювання значення, а не в момент оголошення змінної, таким чином, у різних ділянках програми ті самі змінні може надаватися значення різних типів.

Приклади мов, де є динамічна типізація — Smalltalk, Python, Objective-C, Ruby, PHP, Perl, Javascript, Lisp, xbase.

Прототипне програмування — стиль об'єктно-орієнтованого програмування, при якому відсутнє поняття класу, а повторне використання (спадкування) проводиться шляхом клонування існуючого екземпляра об'єкта — прототипу.

Основні властивості

Javascript не потрібно компілювати, він підключається до HTML-сторінки й працює "як є".

Javascript може:

- змінювати сторінку,
- писати на ній текст,
- додавати й видаляти теги,
- міняти стилі елементів.
- реагувати на події: (скрипт може чекати, коли що-небудь трапиться (клік миші, закінчення завантаження сторінки) і реагувати на це виконанням функції.)
 - виконувати запити до сервера й завантажувати дані без перезавантаження сторінки. (Це іноді називають "AJAX".)
 - встановлювати й зчитувати cookie,
 - валідувати дані (перевіряти їх коректність),
 - виводити повідомлення (й багато чого іншого.)

Унікальність javascript:

- Повна інтеграція із браузером

- Прості речі робляться просто
- Підтримується майже скрізь

Структура мови

Структурно Javascript можна представити у вигляді об'єднання трьох чітко помітних друг від друга частин:

- ядро (EcmaScript),
- об'єктна модель браузера (Browser Object Model або BOM),
- об'єктна модель документа (Document Object Model або DOM).

Якщо розглядати Javascript у відмінних від браузера застосуваннях, то об'єктна модель браузера й об'єктна модель документа можуть не підтримуватися.

Типи даних

Стрічкові- String - визначається подвійними або одинарними лапками й використовується для символічних даних,

Логічні - Boolean- визначається відсутністю лапок і використовується для значень true=1 або false=0,

Цифрові - Number- визначається відсутністю лапок і використовується для чисел (не символів).

Також існують спеціальні типи даних:

null - відсутність даних,

об'єкт (object) - програмний об'єкт (посилання на нього),

функція (function) - визначення функції.

Змінні

Змінні в javascript слабко типізовані. Це означає, що при оголошенні не потрібно вказувати тип. Можна надати змінній будь-яке значення.

Однак при цьому інтерпретатор javascript (браузер) завжди знає, якого типу значення містить дана змінна, а значить - які операції до неї застосовні.

Змінну не обов'язково оголошувати. Досить їй просто надати значення: `x = 5`

В іменах змінних можуть використовуватися латинські букви (a...z, A...Z), цифри (0...9), знак долара (\$) і знак підкреслення (_), при цьому не можна використовувати цифру першою.

Бібліотека JQuery

З розвитком технологій функціональність веб-сторінок постійно зростає й наближається до функціональності настільних прикладних програм. Ця зростаюча функціональність реалізується за допомогою Javascript. І зовсім не обов'язково особисто прописувати всі необхідні ефекти, якщо для цього вже написано кілька десятків або навіть сотень бібліотек, що дозволяють реалізовувати ці ефекти. І серед безлічі цих бібліотек по праву найбільш зручною й доступною для розуміння є **бібліотека jquery**.

Давайте розберемося. jquery - бібліотека Javascript, що фокусується на взаємодії Javascript і HTML. Була опублікована на комп'ютерній конференції «Barcamp» у Нью-Йорку Джоном Ресигом в 2006 році. У чому головна перевага jquery? Вона закладена на рівні ядра - це вибір елементів об'єктної

моделі документів. Крім того, завдяки наявності плагінів, базова функціональність jquery може бути розширена.

Для початку роботи з jquery необхідно скачати саму бібліотеку з будь-якого доступного джерела. Далі її необхідно ініціалізувати. Зробити це просто - необхідно додати в заголовок html-документа наступний шматочок програмного коду:

```
<script type="text/javascript" src="/jquery.js"></script>, де jquery.js - ім'я файлу, що містить бібліотеку jquery.  
А щоб jquery-скрипт виконувався разом із завантаженням документа, нижче описаний приклад коду:  
$(document).ready(function(){  
  //Code here  
});, де code here - це код виконуваної функції.
```

Тепер, після ініціалізації, для використання доступні всі можливості базового функціонала jquery, серед яких:

- функції ядра;
- робота із селекторами;
- робота з атрибутами;
- обхід дерева DOM;
- маніпуляції елементами;
- робота з CSS-властивостями елементів;
- робота з подіями;
- візуальні ефекти;
- взаємодія з ажах;
- утиліти.

Для маніпулювання потрібними елементами сторінки в Javascript є кілька способів знайти їх на сторінці серед безлічі інших об'єктів. Ці способи вимагають запам'ятовування великої кількості інформації, у той час як для пошуку елемента за допомогою jquery необхідно лише пам'ятати ID елемента, з яким ви прагнете працювати. Код звернення в загальному випадку буде виглядати так:

`$('.шлях до елемента (елементам)')`

Ключовою функцією в jquery є функція `$()` - вона тим чи іншим способом викликається всіма методами jquery.

Згідно із заявами вице-президента по розробці Скотта Гутрі (Майкрософт), бібліотека jquery, стане основою **ASP.Net Ajax Control Toolkit** і буде поставлятися в складі Visual Studio. Розроблювач jquery Джон Резиг заявив, що Нокія теж використовує jquery як частину своєї платформи для розробки Web Runtime, що базується на Webkit. Не відстають і російські компанії, наприклад, Яндекс уже давно активно використовує jquery у своїх додатках.

Завдяки тому, що обсяг програмного коду jquery менше, ніж обсяг стандартного коду Javascript, скорочуються часові витрати на розробку елементів веб-сторінки. Сам програмний код більш зрозумілий у порівнянні з Javascript. Наведемо приклад. Існує якась таблиця, непарні рядки якої пофарбовані відмінним від основного кольором. Припустимо, що існує також якась таблиця стилів CSS, у якій уже визначений клас (у прикладі, "odd") для такого стилю. Для реалізації цього ефекту необхідно наділити непарні рядки таблиці даним класом. У реалізації на чистому Javascript програмний код буде виглядати так:

```
var tables = document.getElementsByTagName("table");
for ( var t = 0; t < tables.length; t++ ) {
  var rows = tables[t].getElementsByTagName("tr");
  for ( var i = 1; i < rows.length; i += 2 )
  if ( !/(|s)odd(s|$)/.test( rows[i].classname ) )
  rows[i].classname += " odd";
}
```

Із застосуванням jquery реалізація матиме наступний вигляд:

`$('.tr:nth-child(odd)').addClass("odd");`

Як показує даний приклад, реалізація ефекту за допомогою jquery виглядає більш раціонально в порівнянні з Javascript. Велика кількість плагінів дозволяє реалізувати практично будь-який ефект або дію.

Для створення простого Ajax-Запиту потрібно всього близько п'яти рядків коду, що значно спрощує використання цієї технології

Обробка подій

Важливою ознакою інтерактивних HTML-сторінок є можливість реакції на дії користувача. Наприклад, натиск на кнопці повинен викликати появу діалогового вікна, або виконання перевірки правильності введених користувачем даних. В JavaScript інтерактивність реалізована за допомогою **перехвату** та **обробки подій**, викликаних в результаті дій користувача. Для цього в теги деяких елементів введені параметри **обробки подій**. Ім'я параметру обробки події починається з префіксу `on`, за яким йде назва події. Наприклад, події клік кнопкою миші `Click`, відповідає параметр обробки події з назвою `onClick`. Назви та характеристики деяких подій наведені в табл. 7.1.

Таблиця 7.1 Події JavaScript

Подія	Характеристика події	Обробник події
Click	Клік кнопкою миші на елементі форми або гіперпосилання	onClick
KeyDown	Натиск на клавіші клавіатури	onKeyDown
Load	Завантажується документ в браузер	onLoad
MouseDown	Натиск на кнопці миші	onMouseDown
MouseOver	Курсор знаходиться над елементом	onMouseOver
MouseOut	Курсор покидає зону над елементом	onMouseOut

Задача. Необхідно, щоб при наведенні курсору на комірку таблиці із написом "Привіт" з'являлось вікно повідомлення з фразою "Hello". Можливі рішення:

Варіант 1:

```
<td onClick="alert('Hello')"> Привіт </td>
```

Варіант 2:

```
<script>  
function Go() {  
  alert("Hello")  
}  
</script>  
<td onClick="Go()"> Привіт </td>
```

В варіанті вирішення 1, код JavaScript був записаний безпосередньо в тезі, а в варіанті 2 наслідком кліку став виклик функції. Варіант 2 слід використовувати, якщо код обробки події великий за обсягом.