

## Лабораторна робота №7

### Доступ до об'єктів і сценаріїв. Доступ до властивостей і методів об'єктів

**Мета роботи** - навчитися використовувати різні способи доступу до властивостей і методів об'єктів для внесення змін в HTML-документ.

Всі зміни в HTML-документі виконуються через властивості, методи і події об'єктів, що входять до складу об'єктної моделі документа. Розглянемо сценарій (приклад 1.1), в якому використовуються типові способи доступу до властивостей і методів об'єктів.

Приклад 1.1.

```
<HTML ID='DOCUM'>
<CENTER>
  <H1 ID='zag'> ВАСЬКА</h1>
  Варіант, сумісний з Mozilla
  <P>
    <FORM NAME='f1'>
      <INPUT TYPE=BUTTON name="bot" onclick=bm() value='Збільшити'>
    </FORM >
    <P> <IMG SRC="VaskaM.jpg" ID='Vas' onclick=bm(>
  </CENTER>
<!-- В скрипті закоментований вивід вікна зі списком об'єктів -->
<SCRIPT>

/*  var msg=''
    for(i=0; i<document.all.length;i++)
      msg+=i + ' ' + document.all[i].tagName + ' ID=' +
document.all[i].id+'\n'
    alert(msg)  */
flag=true
function bm()
{ if(flag)

{ document.images[0].src='VaskaB.jpg' flag=false;
  document.forms[0].bot.value="Зменшити"
  document.forms[0].bot.style.background='red'
  document.all.bot.style.color='black';
}
else
{ document.getElementById("Vas").src='VaskaM.jpg';
  //обращение к кнопке по индексу
  document.forms[0].bot.value="Збільшити"
  document.forms[0].bot.style.background='green'
  document.all.bot.style.color='white'; flag= true
}

}
</script>
</ HTML>
```

Виконати приклад 1.1

У прикладі 1.1 сценарій використовується для заміни фотографії та зміни кольору кнопки і напису на ній. Один одного змінюють маленька і велика фотографії кота Васьки.

Заміна фотографії та зміна виду кнопки в прикладі відбуваються при натисканні мишкою по кнопці або по самій фотографії. У тегах `<INPUT>` і `<IMG>` поміщений атрибут `onclick = bm ()`, за допомогою якого у відповідь на клацання мишкою викликається функція `bm ()`. Такий атрибут називають оброблювачем подій. Файл з фотографією є властивістю `SRC` об'єкта `IMG`. Заміна фотографії робиться в сценарії двома рівнозначними способами:

```
document.images [0] .src = 'VaskaB.jpg'  
document.getElementById ("Vas"). src = 'VaskaM.jpg'
```

У першому варіанті доступ до фотографії відбувається через колекцію `images`, в якій всі розташовані на сторінці фотографії (в прикладі одна фотографія) пронумеровані. Нумерація починається з нуля. У другому варіанті використовується метод `getElementById ()` об'єкта `document`. Параметром в методі `getElementById ()` служить ID (ідентифікатор) об'єкта `IMG`. Весь оператор читається так: "властивості `src` об'єкта, що має ID = `Vas` і входить до складу об'єкта `document`, привласнити значення `VaskaM.jpg`". Сам об'єкт `document` є дочірнім по відношенню до об'єкта `window`. Тому повне звернення до властивості `src` варто було б писати так:

```
window.document.images [0] .src = 'VaskaB.jpg'  
window.document.getElementById ("Vas"). src =  
'VaskaM.jpg'
```

Зазвичай, за винятком тих випадків, коли потрібно скористатися властивостями або методами самого об'єкта `window`, це слово опускають.

Більш складна ієрархія об'єктів спостерігається при зверненні до властивостей кнопки:

```
document.forms [0] .bot.value = "Зменшити"  
document.forms.f1.bot.style.background = 'red'
```

Перший оператор служить для зміни напису на кнопці. Доступ до кнопки відбувається по її імені (`name = 'bot'`). Доступ до форми здійснюється через колекцію `forms` і номер форми в колекції.

За допомогою другого оператора змінюється колір кнопки. Доступ до форми здійснюється через колекцію `forms` і ім'я форми `f1`. Своєрідність доступу до властивостей, що задається стилем, полягає в тому, що `style` виступає формально як дочірній об'єкт. У розглянутому прикладі батьківським по відношенню до `style` є об'єкт з ім'ям `bot`, тобто, кнопка.

Для звернення до об'єктів зручна колекція `all` об'єкта `document`. У колекцію `all` входять всі об'єкти HTML-документа, тому вибирати об'єкт потрібно по його ID або імені. Колекція `all` використана для зміни кольору написи на кнопці:

```
document.all.bot.style.color = 'black'
```

Всі описані способи доступу до об'єктів HTML-документа базуються на об'єктній моделі документа (DOM). Найбільш важким завданням, що стоїть перед розробником сценаріїв, є розробка сайтів, які будуть однаково

виконуватися на всіх браузерях. Розглянутий приклад HTML-документа підібраний так, що він (майже) однаково виконується на браузерах Internet Explorer і Mozilla Firefox, незважаючи на те, що об'єктні моделі в цих браузерах відрізняються одна від одної. Для того, щоб переконатися в цьому приберіть символи коментарів / \* і \* / з першого і п'ятого рядка скрипта і відкрийте сторінку в обох браузерах. З'явиться вікно зі списком усіх об'єктів, створених браузером для відкритого документа. У створеній браузером Internet Explorer моделі - 12 об'єктів, а у Mozilla Firefox - 10.

Розглянемо докладніше.

Розкоментуйте фрагмент скрипта.

```
var msg = ''
for (i = 0; i <document.all.length; i ++)
msg += i + ' ' + document.all [i] .tagName + 'ID =' +
document.all [i] .id + '\ n'
alert (msg)
```

Колекція all всіх об'єктів HTML-документа має властивість length, дорівнює кількості всіх об'єктів моделі документа. Будь-який об'єкт має властивість tagName, значення якого збігається з ім'ям об'єкта. У наведеному фрагменті скрипта в циклі формується строкова змінна msg, що складається з номерів, імен та ID об'єктів. Поєднання символів \ n служить для перекладу рядка.

Метод getElementById () - входить в стандарт W3C і підтримується всіма браузерами. Колекція all розроблена фірмою Microsoft для браузера Internet Explorer. В даний час ця колекція підтримується багатьма браузерами. Як не дивно, в новому браузері Edge, вбудованому в ОС Windows 10, колекція all не підтримується.

### **Завдання 1.1.**

Напишіть HTML-документ, який у вікні браузера відображається у вигляді наступних трьох рядків:

- Доступ до властивостей і методів
- Колекція all
- Метод getElementById ()

Перший рядок помістіть в контейнер <H2> ... </ H2>, другий - в контейнер <P> ... </ P>, третій - в контейнер <DIV> ... </ DIV>. Напишіть скрипт для зміни кольорів фону і літер написів, при натисканні мишкою на ці рядки. При натисканні на першому рядку колір букв повинен змінюватися з чорного на білий або з білого на чорний, а фон - з жовтого на синій або з синього на жовтий. Так само повинні змінюватися кольору третього рядка.

При клацаннях по другому рядку колір букв у ньому повинен змінюватися з червоного на білий і навпаки, а колір фону - з білого на зелений і навпаки.

Щоб змінити перший і третій рядок застосуєте метод getElementById (), а для другого рядка - колекцію all.

### ***Розміщення сценаріїв в HTML-документі***

Сценарій в HTML-документі можна розміщувати трьома способами:

- у відкриваючому тегу як значення атрибуту подія (див. Приклад 2.1);
- у контейнері, обмеженому тегами <SCRIPT> ... </SCRIPT>;
- в окремому файлі.

Контейнери зі сценаріями можуть розміщуватися в будь-якому місці HTML-документа. Кількість сценаріїв в одному HTML-документі не обмежена.

Виконання сценаріїв відбувається при завантаженні HTML-документа і при настанні подій. Під час завантаження HTML-документа браузер послідовно переглядає сценарії і намагається їх виконати. Якщо зустрівся виклик функції, то браузер шукає її опис в поточному і в усіх раніше завантажених сценаріях. Тому опис функції повинен розміщуватись або в одному сценарії з викликом, або в сценаріях розташованих вище виклику функції.

Сценарій можна зберігати в окремому файлі. Для вставки сценарію в HTML-документ служить атрибут SRC тега <SCRIPT>. Сценарій, розміщений в окремому файлі, можна використовувати на багатьох сторінках сайту. У прикладі 1.2 розглядається одна зі сторінок сайту мережі магазинів.

### **Приклад 1.2**

```
<HTML> <! - СКРИПТ завантажується з файлу primJs.js -> <HEAD>
<TITLE> Мережа </TITLE>
</HEAD>
<BODY>
<SCRIPT language = "JavaScript" src = "PrimJs.js">
</SCRIPT>
<H2 align = center style = "color: green"> Магазин "ПОДАРУНКИ"
</H2>
Адреса: Лісова вул., Буд.2 <P>
Транспорт: трамваї 7, 23, автобуси 56, 93
</BODY>
</HTML>
```

```
// Файл primJs.js
a = "background-color: # 00ffff; color: # ff00ff;"
a + = "font-size: 24pt; font-family: 'Times New Roman'"
naim = 'Мережа магазинів "ВСЕ ДЛЯ БУДИНКУ"'
var da = new Date ()
d = da.getDate () + "." + (da.getMonth () + 1) + "." +
da.getFullYear ()
document.write ( '<P align = center style = "' + a + '">' +
naim + '</P> <P> Сьогодні' + d + '</P>')
```

Виконати приклад 1.2

Загальний для сторінок всіх магазинів заголовок з назвою мережі та сьогоднішня дата формуються скриптом, розміщеним в окремому файлі primJs.js. Метод write (HTML-код) служить для вставки в сторінку розміченого

тексту (HTML-коду). Цей метод застосовується тільки при завантаженні сторінки, так як його застосування після закінчення завантаження призведе до стирання старого вмісту сторінки.

**Завдання 1.2.** Додайте в приклад 1.2 три сторінки. Дві сторінки повинні відображати інформацію про магазини Посуд і Меблі. Третя сторінка - головна в сайті мережі магазинів ВСЕ ДЛЯ ДОМУ. На ній повинні бути посилання на сторінки магазинів, що входять в мережу. На сторінках магазинів повинен використовуватися скрипт з файлу `primJs.js`.