

# Мова програмування

## JavaScript

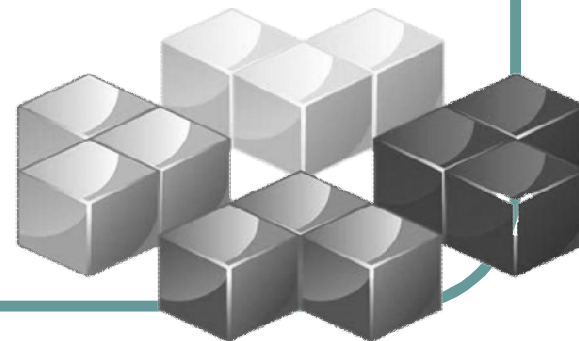
*JAVASCRIPT*

*Огляд;*

*Об'єкти;*

*Події;*

*Функції.*



**JavaScript - мова програмування з об'єктно-орієнтованими можливостями.**

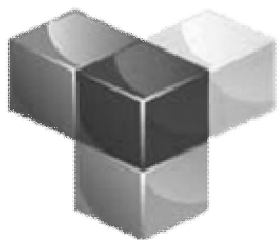
**JavaScript - прототипно-орієнтована скриптова мова програмування**

**JavaScript зазвичай використовується як вбудовувана мова для програмного доступу до об'єктів додатків.**

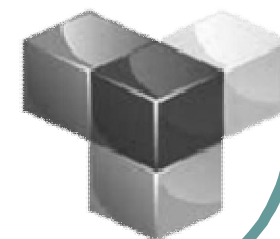
**Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок.**

**Назва «JavaScript» є зареєстрованим товарним знаком компанії Oracle Corporation.**

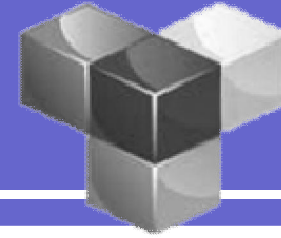
**За допомогою JavaScript Ви зможете створювати інтерактивні веб-сторінки.**



**Інтерактивні сторінки можуть взаємодіяти з користувачем (виводити повідомлення, змінювати вміст після певних дій і т.д.)**



# JavaScript



*JavaScript – це мова призначена для написання скриптів, які виконуються в браузері і вбудовуються в HTML сторінки.*

*JavaScript звичайно вбудовується безпосередньо у HTML документ, додаючи web-сторінці інтерактивності, що робить її привабливою для відвідувачів і зручною для перегляду.*

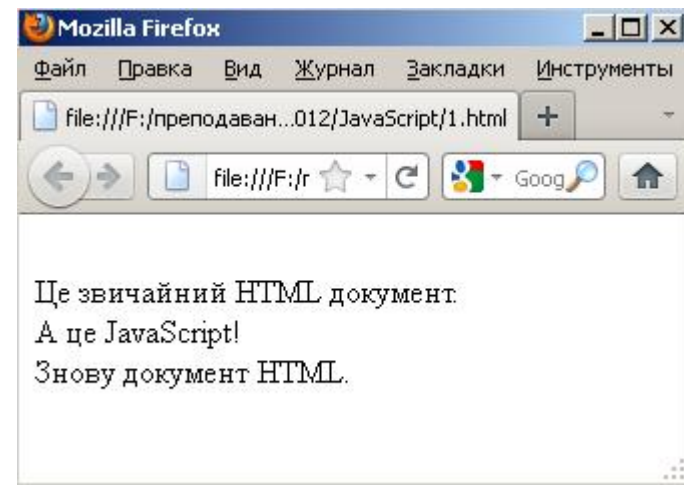
*Для вивчення програмування на мові JavaScript необхідно знати основи HTML.*

*Javascript підключається безпосередньо в HTML-файл. Найпростіший спосіб - це написати javascript-команди всередині тега `<script>` будь де в тілі сторінки.*

# Загальні відомості про JavaScript

*Код скрипту JavaScript розміщується безпосередньо на HTML-сторінці. Щоб побачити, як це робиться, давайте розглянемо наступний простий приклад:*

```
<html>
<body>
<br>
Це звичайний HTML документ.
<br>
  <script language="JavaScript">
    document.write("А це JavaScript!")
  </script>
<br>
Знову документ HTML.
</body>
</html>
```



# Події

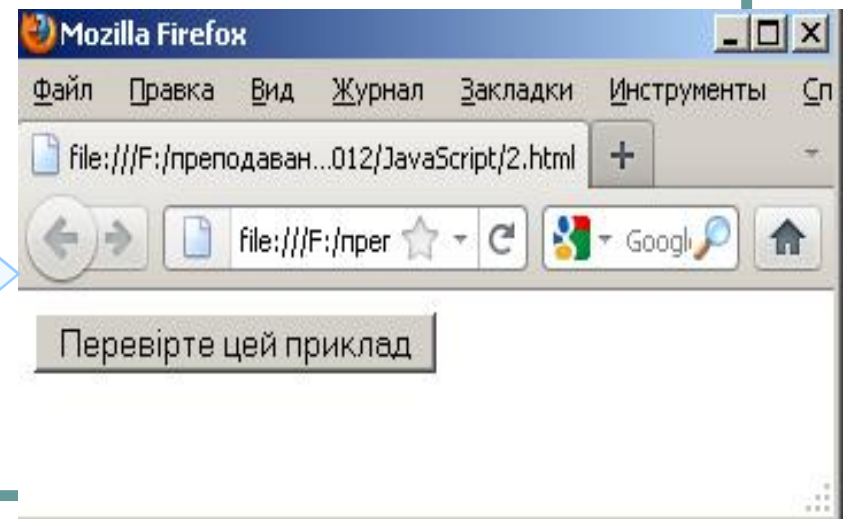
Події, головним чином, ініціюються тими або іншими діями відвідувача.

Якщо клацати по деякій кнопці, відбувається подія *"Click"*.

Якщо покажчик миші перетинає будь-яке посилання гіпертексту - відбувається подія *MouseOver*.

Наведений нижче код представляє простий приклад програми обробки події *onClick*:

```
<form>  
<input type="button" value=  
"Перевірте цей приклад"  
onClick="alert('Це текст')">  
</form>
```



# Базові події JavaScript

Події	Опис
onAbort	Відбувається при перериванні завантаження графічного зображення.
onBlur	Відбувається, коли який-небудь елемент губить фокус.
onChange	Відбувається при зміні значення текстового поля.
onClick	Відбувається при натисканні кнопки миші в області елемента.
onError	Відбувається при помилці завантаження документа або графічного зображення.
onFocus	Відбувається при одержанні елементом фокуса.
onLoad	Відбувається по завершенні завантаження сторінок або графічного зображення.
onMouseOver	Відбувається при переміщенні курсору миші в області елемента.
onMouseOut	Відбувається при переміщенні курсору миші з області елемента.
onReset	Відбувається при натисканні кнопки типу Reset.
onSelect	Відбувається при виборі тексту в текстовому полі.
onSubmit	Відбувається при натисканні кнопки типу Submit
onUnload	Відбувається при переході на іншу сторінку або при завершенні роботи із браузером.

*Розглянемо кілька прикладів які демонструють роботу із цими подіями*

## onLoad

```
<script language="JavaScript">
function doLoad() {
alert("Завантаження зображення завершено")
}
</script>
<body>

</body>
```

## onMouseOver і onMouseOut

```
<script language="JavaScript">
function doMouseOver() {
alert("Покажчик на посиланні")
}
function doMouseOut()
{
alert("Покажчик за межами посилання")
}
</script>
<body>
<a href="index.htm" onMouseOver="doMouseOver"
onMouseOut="doMouseOut()">
Зміст</a>
</body>
```

# Функції

Функції являють собою лише спосіб зв'язати разом декілька команд. Розглянемо простий підхід:

```
<html>
<script language="JavaScript">

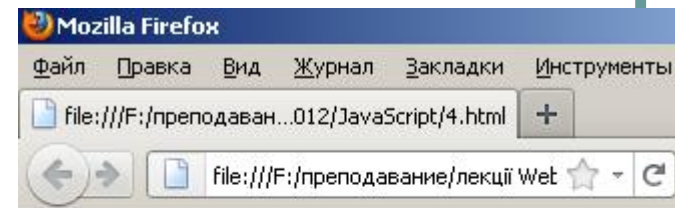
document.write("Ласкаво просимо на мою сторінку!<br>")
document.write("Це JavaScript!<br>")
document.write("Ласкаво просимо на мою сторінку!<br>")
document.write("Це JavaScript!<br>")
document.write("Ласкаво просимо на мою сторінку!<br>")
document.write("Це JavaScript!<br>")
|
</script>
</html>
```

**Приклад скрипту для рішення того ж самого завдання:**

```
<html>
<script language="JavaScript">

function myFunction() {
    document.write("Ласкаво просимо на мою сторінку!<br>")
    document.write("Це JavaScript!<br>")
}
myFunction()
myFunction()
myFunction()

</script>
</html>
```



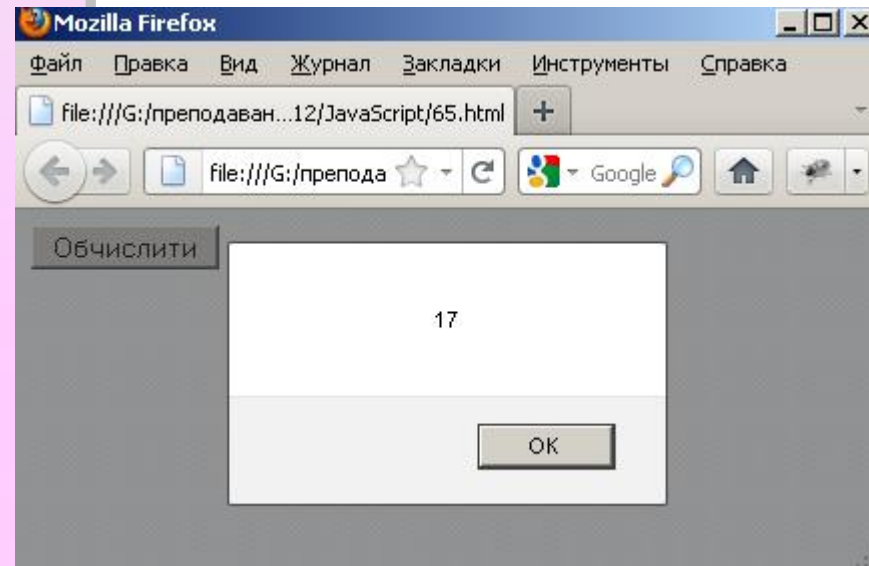
Ласкаво просимо на мою сторінку!  
Це JavaScript!  
Ласкаво просимо на мою сторінку!  
Це JavaScript!  
Ласкаво просимо на мою сторінку!  
Це JavaScript!



Всі команди скрипту, що перебувають усередині фігурних дужок - `{ }` - належать функції *myFunction()*.

Функції можуть також використовуватися разом із процедурами обробки подій.  
Розглянемо наступний приклад:

```
<html>
<head>
<script language="JavaScript">
function calculation() {
  var x= 12
  var y= 5
  var result= x + y
  alert(result)
}
</script>
</head>
<body>
<form>
<input type="button" value="Обчислити"
onClick="calculation()">
</form>
</body>
</html>
```



# Ієрархія об'єктів в JavaScript

*У мові JavaScript всі елементи на web-сторінці вибудовуються в ієрархічну структуру. Кожний елемент з'являється у вигляді об'єкта. І кожний такий об'єкт може мати певні властивості й методи.*

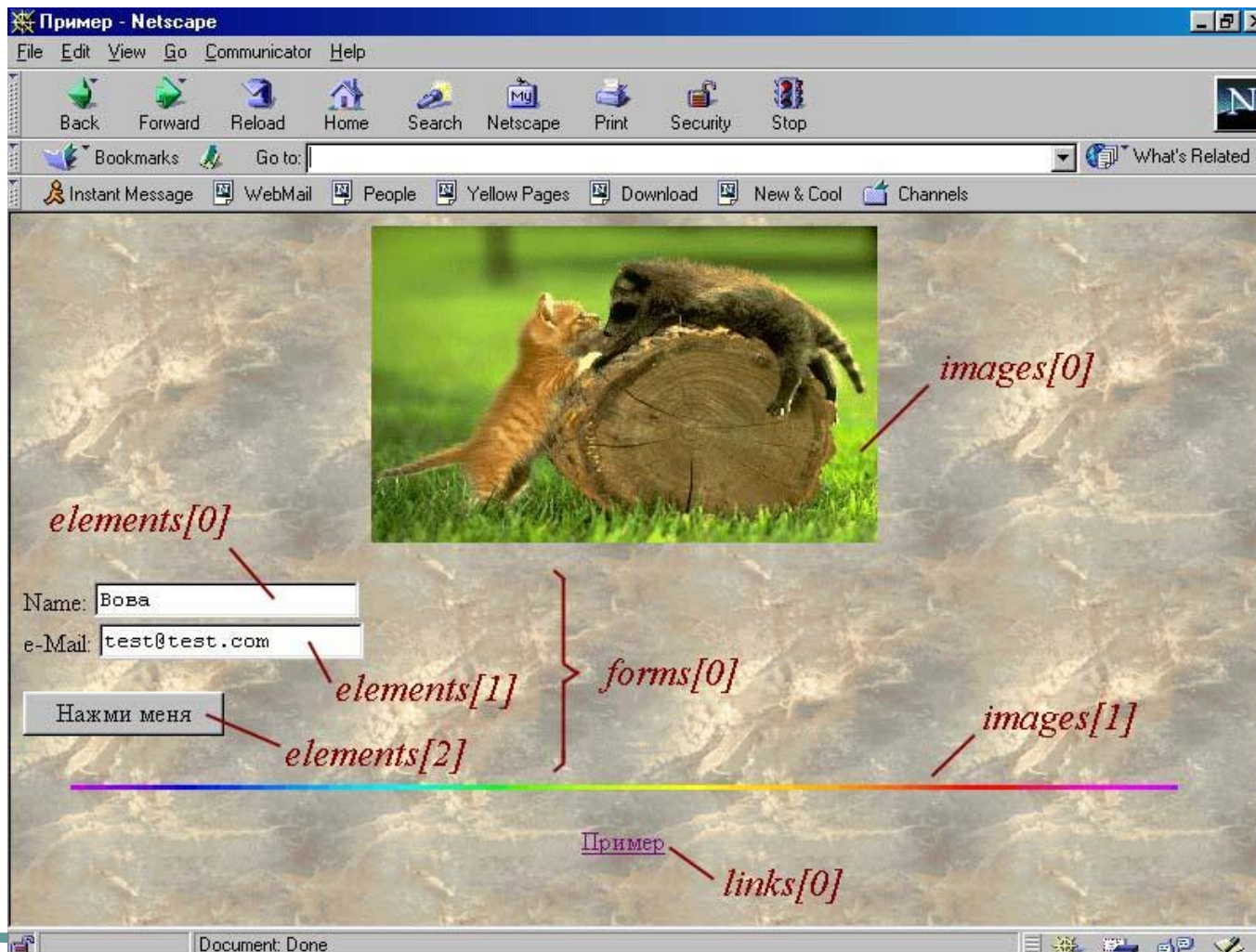
*Розглянемо простий приклад:*

```
1 - Блокнот
Файл  Правка  Формат  Вид  Справка
<html>
<head>
<title>приклад
</head>
<body background="bg.gif" TEXT="#000000" LINK="#B50000" VLINK="#800080" ALINK="#800080">
<center>

</center>
<p>
<form name="myForm">
Name:
<input type="text" name="name" value="Вова"><br>
e-Mail:
<input type="text" name="email" value="test@test.com"><br><br>
<input type="button" value="Натисни мене" name="mybutton" onClick="alert('привіт')">
</form>
</p>
</center>
<p>

</p>
<p>
<a href="1.htm">приклад</a>
</p>
</center>
</body>
</html>
```

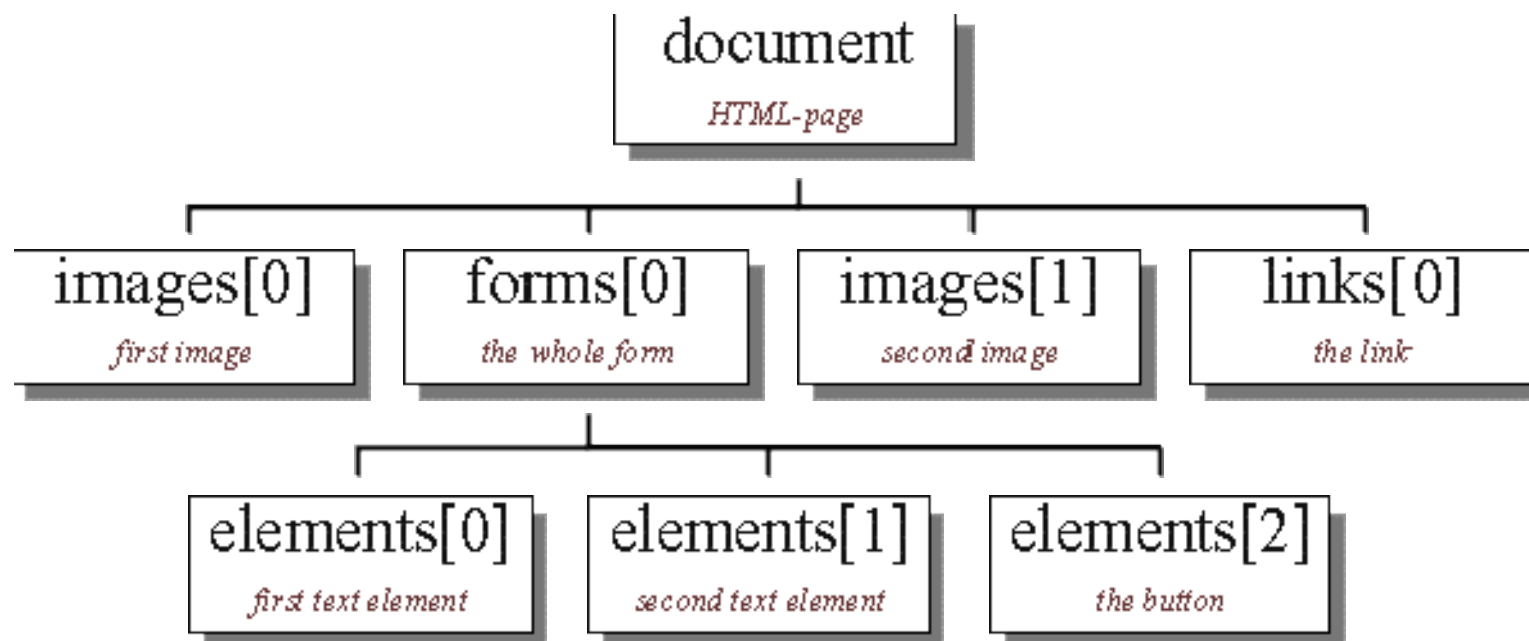
# Вигляд web-сторінки з вище розглянутого прикладу



## Пояснення до поданого прикладу

Отже, ми маємо два малюнки, одне посилання і форму із двома полями для введення тексту й однією кнопкою. З погляду мови JavaScript вікно браузера - це якийсь об'єкт *window*. Цей об'єкт також містить у свою чергу деякі елементи оформлення, такі як рядок стану. Усередині вікна ми можемо розмістити документ HTML. Така сторінка є ні чим іншим, як об'єктом *document*. Це означає, що об'єкт *document* представляє в мові JavaScript завантажений на дійсний момент документ HTML. Об'єкт *document* є дуже важливим об'єктом у мові JavaScript і ви будете користуватися ним багаторазово. До властивостей об'єкта *document* відносяться, наприклад, колір тіла для web-сторінки. Однак для нас набагато важливіше те, що всі без винятку об'єкти HTML є властивостями об'єкта *document*. Прикладами об'єкта HTML є, приміром, посилання або заповнювана форма.

Ієрархія об'єктів, створювана HTML-сторінкою 3  
нашого прикладу:



## Пояснення до поданого прикладу

Для того щоб дізнатися який текст увів користувач у перший елемент форми, то спершу необхідно з'ясувати, як одержати доступ до цього об'єкта. Починаємо ми з вершини нашої ієрархії об'єктів. Потім простежуємо шлях до об'єкта з ім'ям `elements[0]` і послідовно записуємо назви всіх об'єктів, які минемо. У підсумку з'ясовується, що доступ до першого поля для введення тексту можна одержати, записавши: `document.forms[0].elements[0]`.

А тепер як довідатися який текст був введений користувачем? Елемент, що відповідає полю для введення тексту, має властивість `value`, що саме й відповідає введеному тексту. Отже, тепер ми маємо все необхідне, щоб прочитати введене значення. Для цього потрібно написати мовою JavaScript рядок: `name= document.forms[0].elements[0].value`. Отриманий рядок заноситься в змінну `name`. Отже, тепер ми можемо працювати із цієї змінної, як нам необхідно. Наприклад, ми можемо створити вікно, що випадає, скориставшись командою `alert("Привіт" + name)`. У результаті, якщо читач уведе в це поле слово 'Іван', те по команді `alert("Привіт " + name)` буде відкрите вікно, що випадає, з вітанням 'Привіт Іван'.

Також ви можете самі привласнювати різним об'єктам унікальні імена. Як це робиться, ви можете побачити знову ж у нашому прикладі:

```
<form name="myForm">
```

```
Name:
```

```
<input type="text" name="name" value="Іван"><br>
```

```
...
```

Це значно спрощує програмування на JavaScript

# Об'єкт location

Крім об'єктів *window* і *document* в JavaScript є ще один важливий об'єкт - *location*. У цьому об'єкті представлена адреса завантаженого HTML-документа. Наприклад, якщо ви завантажили сторінку *index.htm*, те значення *location.href* саме й буде відповідати цій адресі.

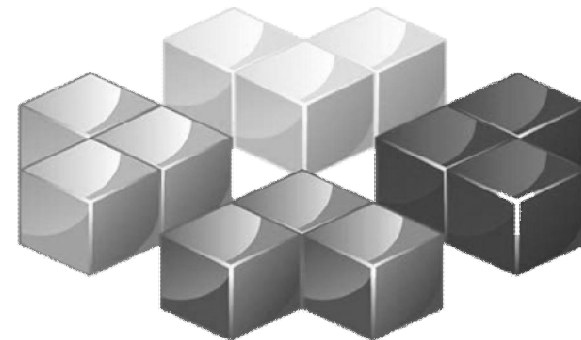
Для нас набагато більш важливо, що ми маємо можливість записувати в *location.href* нові значення. Наприклад, у даному прикладі кнопка завантажує в поточне вікно нову сторінку:

```
<form>  
<input type=button value="До змісту"  
onClick="location.href='index.htm'">  
</form>
```

# Завдання

## Питання для самоконтролю:

1. Які дескриптори використовуються для створення сценарію, форми?
2. Назвіть базові події JavaScript.
3. Синтаксис функцій в JavaScript.
4. Ієрархія об'єктів в JavaScript.
5. Призначення об'єкта Document.
6. Призначення об'єкта Location.



## Завдання на лабораторну роботу

1. Вивчити теоретичний матеріал.
2. Створити Web-сторінку з використанням вивченого матеріалу.
3. Захистити роботу викладачеві.