**JavaScript : không phải là ngôn ngữ Java**

**-JavaScript là điểm sử lý những script ở phía client cho nền tảng web**

**-là ngôn ngữ Client-Side- Script hoạt động trên trình duyệt của người dùng (client)**

**-chia sẽ sử lý trong ứng dụng web .giảm các sử lý không cần thiết trên server.**

**-tạo các hiệu ứng tương tác cho web.**

-Script đặt trong thẻ head thì: script sẽ thực thi ngay khi trang web được mở.

-Script đặt trong thẻ body thì:script trong phần body được thực thi khi trang web đang mở(sau khi thực thi các đoạn script có trong thẻ <head>)

-số lượng đoạn client-script chèn vào trang không hạn chế.

**Biến số trong JavaScript:**

**Cách đặt tên biến:**

-bắt đầu bằng một chữ cái hoặc dấu\_ không thể bằng số

-Phân biệt Hoa,Thường.

**Khái báo biến:**

-Sử dụng từ khoá var

Ví dụ: var count=10,amount;

-Không cần khai báo biến trước khi sử dụng, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu tiên.

-Biến không cần khai báo kiểu dữ liệu vì biến trong javascript không có kiểu dữ liệu nhất định.

**Kiểu dữ liệu trong javascript:**

|  |  |  |
| --- | --- | --- |
| Kiểu dữ liệu | Ví dụ | Mô tả |
| Object | Var listBooks=new Array(10); | Trước khi sử dụng ,phải cấp phát bằng từ khoá new |
| String | “ the abcdsd”  “ ”  “ 40” | Chứa được chuổi unicode  Chuỗi rỗng “” |
| Number | 0.012345  12 | Theo chuẩn IEEE 754 |
| Boolean | True/false | Kiểu nhị phân |
| Undefined | Var myVarible; | myVariable =undefined  -không gán giá trị ban đầu thì tự nhận kiểu dữ liệu undefined :không xác định. |
| Null | Connection.Close(); | Connection=null |

**Đổi kiểu dữ liệu :**

Biến tự đổi kiểu dữ liệu khi giá trị mà ló lưu trữ thay đổi

Ví dụ:

var x=10; // x kiểu Number

var x=”hello world”; //x kiểu String

có thể cộng 2 biến khác kiểu dữ liệu:

ví dụ:

var x;

x=”12” + 34.5; javascript ưu tiên kiểu dữ liệu đầu tiên.

//kq: x=”1234.5”

Sài Hàm parseInt( ), parseFloat( ):đổi kiểu dữ liệu từ chuỗi sang số.

**Hàm trong JavaScript :**

**Dạng thức khai báo chung :**

function Tên\_hàm(thamso1,thamso2,…)

{

….

}

**Hàm có giá trị trả về:**

function Tên\_hàm(thamso1,thamso2,…)

{

….

return(value);

}

Ví dụ:

function Sum(x,y)

{

tong=x+y;

return tong;

}

Gọi hàm var x =Sum(10,20);

alert(x); //xuất thông báo :30

**document.write(“ ”);** hàm dùng để viết ra màn hình

**một số sự kiện thông dụng trong javascript:**

onClick:

onFocus:

onChange: khi thay đổi

onBlur: đem chuột ra khỏi thẻ đó

onMouseOver :rê chuột vào.

onMouseOut :rê chuột ra.

onMouseDown: khi nhấn giữ.

onMouseUp: nhã ra

onLoad: load form.

onSubmit :khi gửi

var theP=document.getElementById("idxuat");

var dropdown=document.getElementById("idchon");

theP.innerHTML =dropdown.value; //gáng giá trị của dropdown vào thẻ p

**DOM trong javascript**

**Đối tượng window :**

-là thể hiện của đối tượng cửa sổ trình duyệt

-tồn tại khi mở một tài liệu html

-sử dụng đê truy cập thông tin của các đối tượng trên cửa sổ trình duyệt (tên trình duyệt, phiên bản trình duyệt ,thanh tiêu đề ,thanh trạng thái ..)

|  |  |
| --- | --- |
| **Properties** | **Methods** |
| document | Alert |
| event | Confirm |
| history | Prompt |
| location | Blur |
| name | close |
| navigator | Focus |
| screen | Open |
| status |  |

**Đối tượng Document**

-biểu diễn cho nội dung html đang được hiển thị

-dùng để lấy thông tin về tài liệu, các thành phần html và xử lý sự kiện

|  |  |
| --- | --- |
| **Properties** | **Methods** |
| AlinkColor | Close |
| bgColor | Open |
| Body | createTextNode(“”) |
| fgColor | createElement(“HTMLtag”) |
| linkColor | getElementById(“id) |
| Title |  |
| URL |  |
| vlinkColor |  |
| Forms[] |  |
| Images[] |  |
| childNodes[] |  |
| documentElement |  |
| cookie |  |

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| document.getElementById(id) | Tìm thẻ thông qua id của thẻ |
| Document.getElement**s**ByTagName(name) | Tìm tất cả các thẻ thông qua tên thẻ |
| Document.getElement**s**ByClassName(name) | Tìm tất cả các thẻ thông qua Css class |
| Document.createElement(element) | Tạo một thẻ mới |
| Document.removeChild(element) | Xoá một thẻ |
| Document.appendChild(element) | Thêm một thẻ |
| Document.replaceChild(element) | Thay thế một thẻ |
| Document.write(text) | Viết ngay vào Html |

**Đối tượng Date trong javascript:**

Var d= new Date();

|  |  |
| --- | --- |
| Method | Mô tả |
| d.getDate(); | Lấy ngày (1-31) |
| d.getDay(); | Lấy ngày trong tuần (0-6) |
| d.getFullYear(); | Lấy năm đầy đủ YYYY |
| d.getYear(); | Lấy 2 số cuối năm YY |
| d.getHours(); | Lấy số giờ (0-23) |
| d.getMilliseconds(); | Lấy số mili giây (0-999) |
| d.getMinutes(); | Lấy số phút (0-59) |
| d.getMonth(); | Lấy tháng **(0-11)** |
| d.getSeconds(); | Lấy số giây(0-59) |
| d.getTime(); | Thời gian đã được chuyển đổi sangdạng mili giây |

**Phương thức tìm kiếm chuỗi con :**

indexOf(str) tìm vị trí str xuất hiện đầu tiên

lastIndexOf(str) tìm vị trí str xuất hiện cuối cùng

search()

-để tìm kiếm chuỗi con chúng ta sử dụng phương thức **String.indexOf(str) ,**trong đó str là chuỗi con ,String là chuỗi cha.

Ví dụ :

Var chuoi=”kha dep trai ”;

chuoi.indexOf(“kha”);

-phương thức này trả về kết quả là vị trí xuất hiện đầu tiên của chuỗi (Bắt đầu là vị trí 0), nếu không tìm thấy chuỗi con thì nó sẽ trả về -1.

**Phương thức cắt chuỗi :**

Slice(start,end);

Nếu tham số truyền vào là âm thì nói sẽ tinh ngược lại nghĩa là nó sẽ tính từ cuối lên .

Nếu bạn chỉ truyền một tham số đầu tiên thì nó sẽ tự hiểu vị trí end là vị trí cuối cùng.

Substring(start,end);

Substr(start,length);

Note tất cả các chuỗi đều bắt đầu từ số 0, vì vậy khi tính toán vị trí phải thật cẩn thận

Chuỗi thay thế replace(str\_find,str\_replace)

toUpperCase(): chuyển sang chữ hoa

toLowerCase(): chuyển sang chữ thường

nối chuỗi sử dụng dấu “ + ”

hoặc dùng hàm concat(“kha”,” đẹp trai” );

kết quả : kha đẹp trai

**hộp thoại** :

alert Box : hộp thoại cảnh báo

comfirm box : hộp thoại xác nhận

prompt box : hộp thoại nhập liệu