Fpt University

FPT POLYTECHNIC

PHẦN 1

LẬP TRÌNH JAVASCRIPT

BÀI 4: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VÀ MÔ HÌNH BOM

www.poly.edu.vn

HỆ THỐNG BÀI CŨ

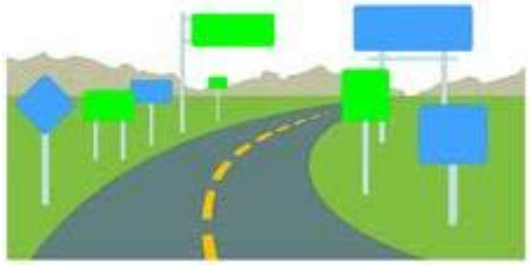
- Cấu trúc điều khiển
- Cấu trúc lựa chọn
  - Lệnh lựa chọn đơn
  - Lệnh lựa chọn kép
  - Lệnh đa lựa chọn
- Cấu trúc lặp
  - Lặp không biết trước số lần lặp
  - Lặp biết trước số lần lặp
- Hàm

Mục tiêu

- Phương thức lập trình
- Phương thức lập trình hướng đối tượng
  - Khái niệm đối tượng, thuộc tính và phương thức
  - Tạo đối tượng
  - Thêm thuộc tính và phương thức vào đối tượng
  - Khái niệm về lớp
  - Định nghĩa lớp, tạo đối tượng từ lớp
  - Các thao tác với đối tượng trong lớp
- Browser Object Model

PHƯƠNG THỨC LẬP TRÌNH

- Lập trình là để giải quyết các vấn đề trong cuộc sống
  - Bài toán tính toán phức tạp: Lập trình cho tên lửa bay vào vũ trụ
  - Bài toán lôgíc: Đưa ra quyết định (dự báo thời tiết)
  - Bài toán quản lý trong các doanh nghiệp (phần mềm tính lương)


- Phương thức lập trình (programing paradigm) đặc tả cách thức giải quyết vấn đề

PHƯƠNG THỨC LẬP TRÌNH

- Có hơn 25 phương thức lập trình
  - Mỗi phương thức lập trình giải quyết cho một vấn đề
  - Một số phương thức khó áp dụng trong thực tiễn lập trình
  - Một số phương thức lập trình chỉ được hưởng ứng bởi một nhóm người hay trong một thời gian ngắn
- Những phương thức lập trình phổ biến:
  - Lập trình hướng sự kiện
  - Lập trình hướng thành phần
  - Lập trình cấu trúc
  - Lập trình hướng đối tượng
- Phương thức lập trình hướng đối tượng được phát triển rộng rãi hơn cả

- Đối tượng là tất cả mọi thứ trong cuộc sống (các đồ vật, sự vật)
  - Ví dụ đối tượng: Quả bóng, cái bàn, ôtô, bông hoa, con người, nhà máy…
- Mỗi đối tượng có đặc tính và hành động riêng
- Ý tưởng chủ đạo của phương thức lập trình hướng đối tượng: Mô phỏng cuộc sống thực trong lập trình
  - Trong cuộc sống có những đối tượng như quả bóng, cái bàn… với các đặc tính và hành động riêng thì lập trình mô phỏng các đối tượng đó với các đặc tính và hành động như thế

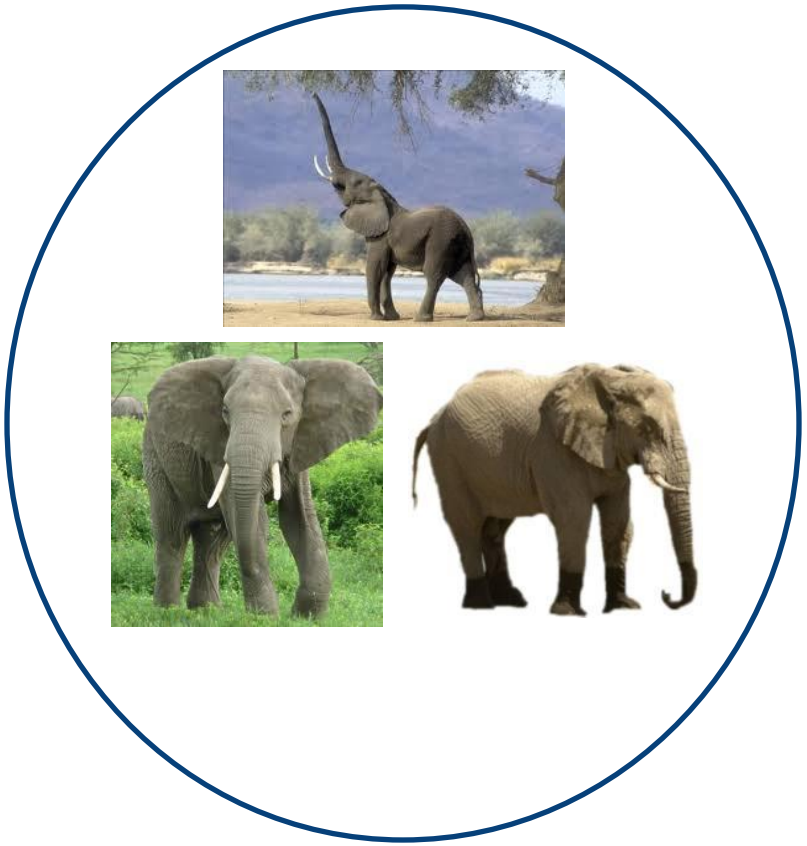
THUỘC TÍNH VÀ PHƯƠNG THỨC

- Trong lập trình: đặc tính được gọi là thuộc tính, hành động được gọi là phương thức

- Mèo có những đặc tính:
  - Màu lông: tam thể
  - Nặng: 2 kg
  - Móng: sắc
- Mèo có những hành động:
  - Bắt chuột
  - Liếm lông

- Voi có những đặc tính:
  - Màu da: nâu
  - Nặng: 2 tấn
  - Vòi: 1m
- Voi có những hành động:
  - Phun nước
  - Ăn cỏ

- Các đối tượng có cùng thuộc tính và phương thức được gom lại thành một lớp
- Hay: Lớp định nghĩa tập hợp các đối tượng có cùng thuộc tính và phương thức



- Sử dụng từ khóa new

```
var tendoituong = new Object();

var meo = new Object();
var hoa = new Object();
```

THÊM VÀ TRUY CẬP ĐẾN THUỘC TÍNH

- Thêm thuộc tính cho đối tượng

tendoituong.tenthuoctinh = giatri;

```
hoaDao.mau = "Hong“;
hoaDao.soCanh = 5;
```

- Truy cập đến thuộc tính của đối tượng

tendoituong.tenthuoctinh

```
alert(hoaDao.mau);
alert(hoaDao.soCanh);
```

THÊM VÀ GỌI PHƯƠNG THỨC

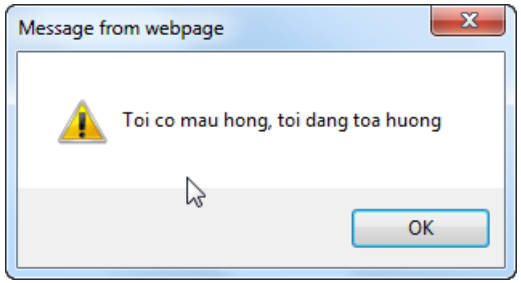
- Thêm phương thức

```
tendoituong.tenphuongthuc = function(){
        //Viết mã cho phương thức ở đây
}
hoaDao.toaHuong= function () {
        alert("Toi co mau Hong, toi dang toa huong");
}
```

- Gọi phương thức

```
tendoituong.tenphuongthuc()

hoaDao.toaHuong();
```



TẠO MỘT KHUÔN MẪU CHO ĐỐI TƯỢNG

- Vấn đề nảy sinh: Giả sử trong vườn có nhiều loại hoa, mỗi loại hoa có màu sắc, số cánh khác nhau

```
var hoaDao = new Object;
hoaDao.mau = "Hong“;
hoaDao.soCanh = "5“;
hoaDao.toaHuong= function () {
        alert("Toi co mau Hong, toi dang
toa huong!");
}
var hoaHong = new Object;
hoaHong.mau = " Do“;
hoaHong.soCanh = " 10“;
hoaHong.toaHuong= function () {
        alert("Toi co mau Do, toi dang
toa huong!");
}
```

```
var hoaCuc = new Object;
hoaCuc.mau = " Vang“;
hoaCuc.soCanh = " 20“;
hoaCuc.toaHuong= function () {
        alert("Toi co mau vang, toi dang
toa huong!");
}
var hoaLan = new Object;
hoaLan.mau = "Tim“;
hoaLan.soCanh = "3“;
hoaLan.toaHuong= function () {
        alert("Toi co mau tim, toi dang
toa huong!");
}
```

- ➔ Tạo một khuôn mẫu chung (lớp)

ĐỊNH NGHĨA LỚP CHO ĐỐI TƯỢNG

- Định nghĩa lớp

```
function tenlop (tenbien1, tenbien2…){
        tenthuoctinh1 = tenbien1;
        tenthuoctinh2 = tenbien2;

        tenphuongthuc = function(){
                //Viết mã cho phương thức ở đây
        }
}

function Hoa(mauHoa, soCanhHoa){
        this.mau = mauHoa;
        this.soCanh = soCanhHoa;
        this.toaHuong = function(){
            alert("toi co mau " + this.mau + ", toi dang toa huong");
         }
}
```

TẠO ĐỐI TƯỢNG VÀ SỬ DỤNG ĐỐI TƯỢNG

- Tạo đối tượng

tenlop["tendoituong"] = new tenlop (giatri1, giatri2…)

```
Hoa ["Dao"] = new Hoa ("Hong", "5");
Hoa ["Hong"] = new Hoa ("Do", "10");
Hoa ["Cuc"] = new Hoa ("Vang", "20");
Hoa ["Lan"] = new Hoa ("Tim", "3");
```

- Truy cập đến thuộc tính và phương thức của đối tượng

tenlop["tendoituong"].tenthuoctinh
tenlop["tendoituong"].tenphuongthuc

```
Hoa ["Dao"].toaHuong();
Hoa ["Hong"].toaHuong();
Hoa ["Cuc"].toaHuong();
Hoa ["Lan"].toaHuong();
```

THAO TÁC VỚI ĐỐI TƯỢNG TRONG LỚP

- Lặp qua các đối tượng

```
for (var x in Hoa) {
        Hoa[x].toaHuong();
}
```

- Tìm một đối tượng trong thuộc lớp

```
if ("Dao" in Hoa) {
        alert(" Doi tuong Dao da duoc tao");
} else {
        alert(" Đối tượng Dao chua duoc tao");
}
```

- Thêm thuộc tính cho đối tượng thuộc lớp

```
if ("Dao" in Hoa) {
        Hoa["Dao"].bieuTuong = "Mua Xuan“;
}
alert("Hoa dao bieu tuong cho " + Hoa["Dao"].bieuTuong);
```

Fpt University

FPT POLYTECHNIC

PHẦN 2

LẬP TRÌNH JAVASCRIPT

BÀI 4: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VÀ MÔ HÌNH BOM

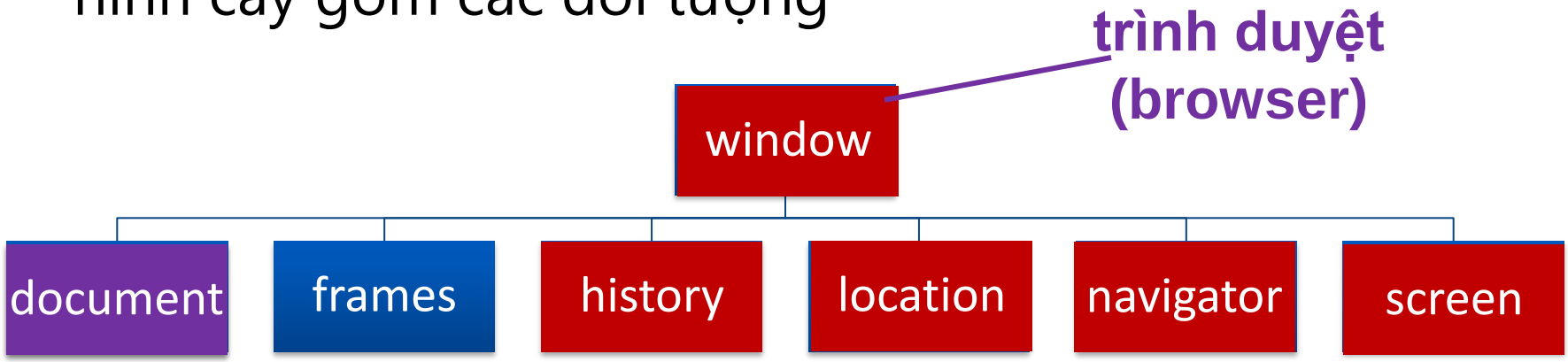
www.poly.edu.vn

BOM

Browser Object Model

BROWSER OBJECT MODEL

- Browser Object Model là một hệ thống phân cấp hình cây gồm các đối tượng



- Các đối tượng cung cấp thuộc tính và phương thức cho lập trình viên JavaScript
- Đối với mỗi đối tượng, mỗi trình duyệt hỗ trợ các thuộc tính và phương thức khác nhau
  - Hiểu môi trường mà trình duyệt cung cấp để viết mã JavaScript chạy ổn định trên nhiều trình duyệt

Đối tượng Window

- Window là đối tượng thể hiển cửa số hiển thị hiện tại trên trình duyệt
- Một số phương thức của đối tượng window đã được sử dụng: alert(), prompt(), confirm()
- Các thuộc tính và phương thức của window có thể gọi trực tiếp hoặc thông qua window

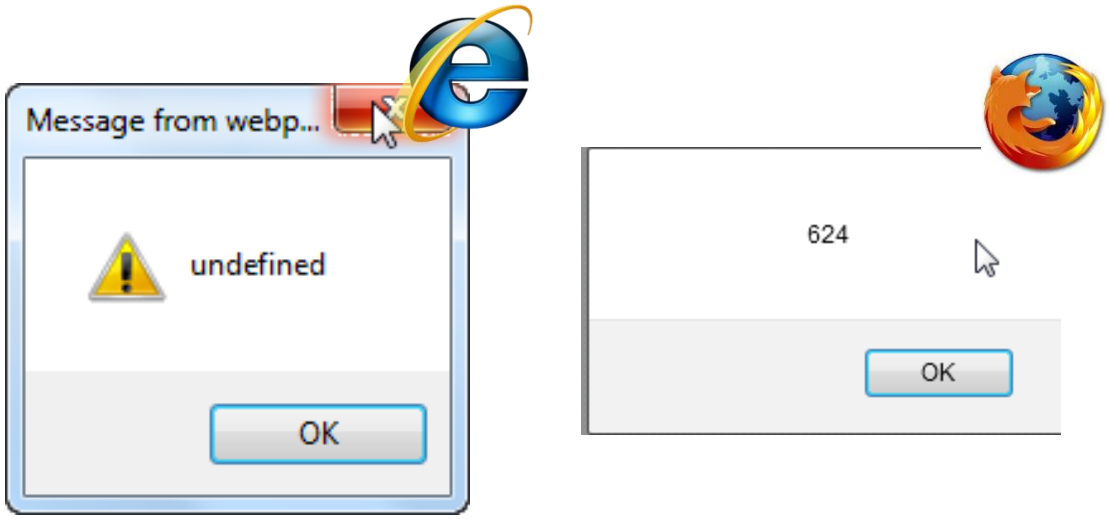
```
alert("Hi")
hoặc
window.alert("Hi")
```

CÁC THUỘC TÍNH CỦA WINDOW

Thuộc tính	Giải thích
closed	Có giá trị là True khi cửa sổ được đóng
defaultStatus	Thiết lập văn bản mặc định trên thanh trạng thái của trình duyệt
name	Thiết lập hoặc trả về tên của cửa sổ
opener	Tham chiếu đến cửa sổ tạo ra cửa sổ hiện tại
status	Thông tin xuất hiện trên thanh trạng thái
innerHeight	Thiết lập hoặc trả về chiều cao phần nội dung của cửa sổ
document	Trả về đối tượng document của cửa sổ

DEMO SỬ DỤNG THUỘC TÍNH INNERHEIGHT

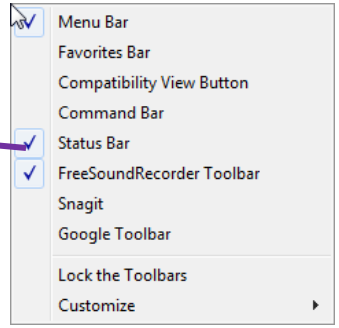
```
alert(window.innerHeight);
```



- FireFox hỗ trợ thuộc tính này trong khi IE không hỗ trợ
- Tham khảo trang w3school để biết được trình duyệt nào hỗ trợ phương thức và thuộc tính nào

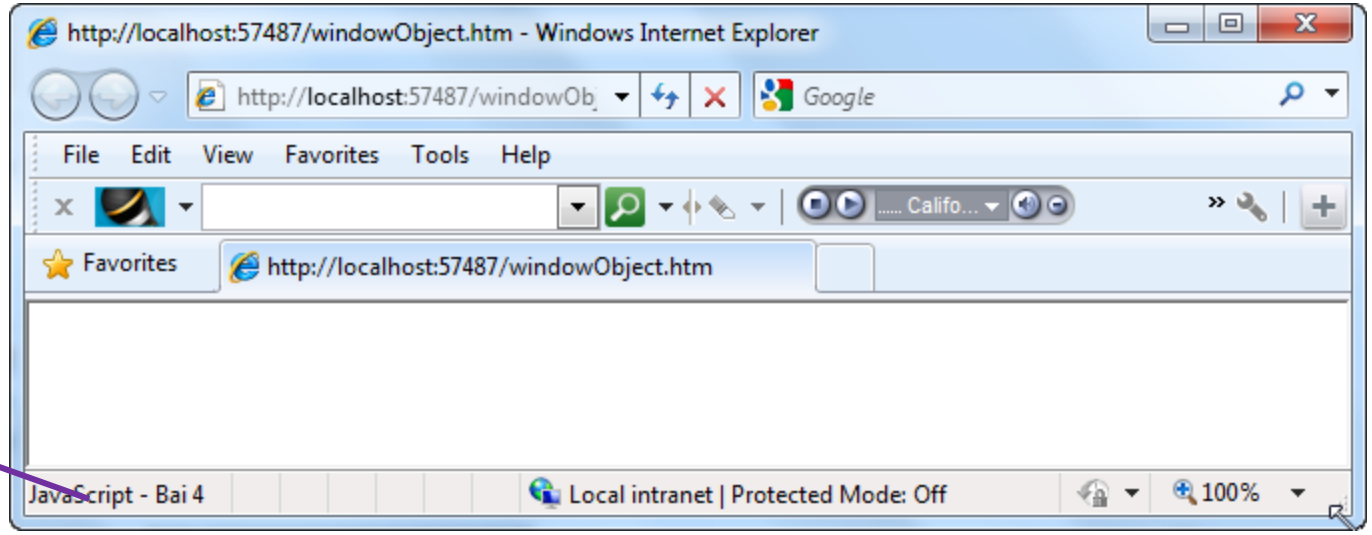
THUỘC TÍNH DEFAULTSTATUS

- Đối với IE, để hiển thị Status bar phải chọn "Status bar"



Chọn Status bar

```
window.defaultStatus= "JavaScript - Bai 4"
```



Status bar

- Phiên bản 8 của Firefox không hỗ trợ default status bar

CÁC PHƯƠNG THỨC CỦA WINDOW

Phương thức	Giải thích
focus()	**Chuyển focus** đến cửa sổ
blur()	**Bỏ focus** đến cửa sổ
close()	**Đóng** cửa sổ
open()	**Mở** cửa sổ
print()	Thực hiện chức năng **in**
moveTo()	Sử dụng để **chuyển cửa sổ về vị trí** xác định
resizeTo()	**Thay đổi kích thước** cửa cửa sổ về vị trí xác định

WINDOW.OPEN

- Sử dụng để mở một cửa sổ từ cửa sổ hiện thời

  window.open(url, ten, dactinh)

  - url: url của trang web
  - ten: tên của cửa sổ sẽ mở
  - dactinh: các đặc tính mà cửa sổ được mở sẽ có (mỗi trình duyệt sẽ hỗ trợ một tập các đặc tính riêng)

  ```
  window.open("http://www.google.com.vn/", "timkiem",
  "menubar = yes, width = 800, height = 600")
  ```

Chú ý:
- Chỉ nên sử dụng cách này khi thật cần thiết vì trình duyệt có thể bị disable javascript
- Có thể sử dụng thẻ <a> để thay thế

ĐỐI TƯỢNG SCREEN

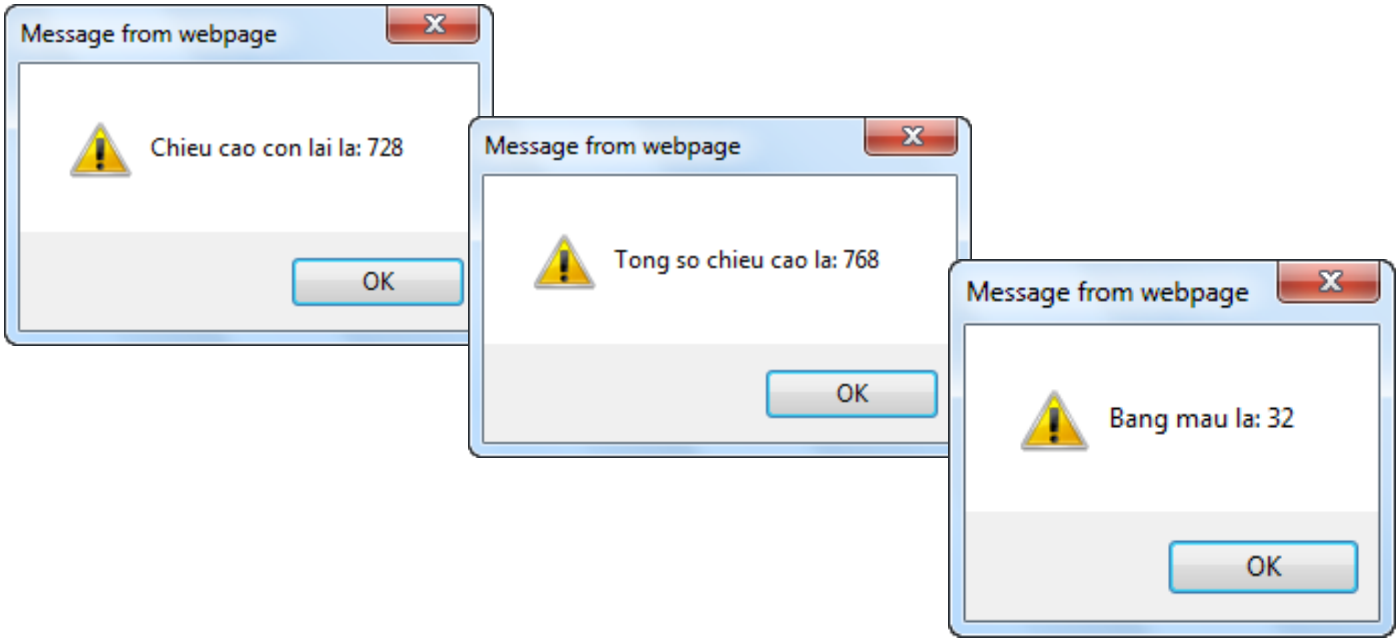
- Mỗi người truy cập sử dụng màn hình có độ phân giải khác nhau, kích thước khác nhau, dải màu khác nhau…
- ➔ Người lập trình phải nắm được thông tin này để hiển thị ảnh phù hợp, hiển thị trang web có kích thước phù hợp…
- Đối tượng screen cung cấp thuộc tính để lấy thông tin về màn hình của người truy cập

ĐỐI TƯỢNG SCREEN

Thuộc tính	Giải thích
availHeight	Trả về **chiều dài** của màn hình (trừ kích thước của window taskbar)
availWidth	Trả về **chiều rộng** của màn hình (trừ kích thước của window taskbar)
height	Trả về **chiều dài** của màn hình
with	Trả về **chiều rộng** của màn hình
pixelDepth	Trả về **độ phân giải** của màn hình
colorDepth	Trả về **bảng màu** để hiển thị ảnh

DEMO THUỘC TÍNH CỦA MÀN HÌNH

```
alert("Chieu cao con lai la: " + screen.availHeight);
alert("Tong so chieu cao la: " + screen.height);
alert("Bang mau la: " + screen.colorDepth);
```



ĐỐI TƯỢNG NAVIGATOR

- Mỗi trình duyệt có cách thức thi hành mã JavaScript riêng
- Có thể cùng thực hiện một chức năng, nhưng đối với từng trình duyệt, cần phải viết các đoạn mã khác nhau
- ➔ Cần biết thông tin về trình duyệt để viết mã JavaScript phù hợp
- Đối tượng Navigator cung cấp các thông tin về trình duyệt

Thuộc tính

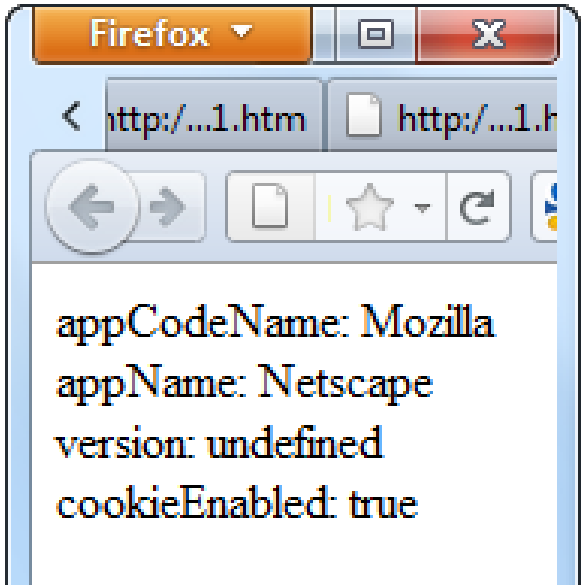
Thuộc tính	Giải thích
appCodeName	Trả về **mã** của trình duyệt
appName	Trả về **tên** của trình duyệt
appVersion	Trả về **phiên bản** của trình duyệt
cookieEnabled	Xác định xem **Cookie** có được bật hay không
platform	Trả về **nền tảng** mà trình duyệt được biên dịch

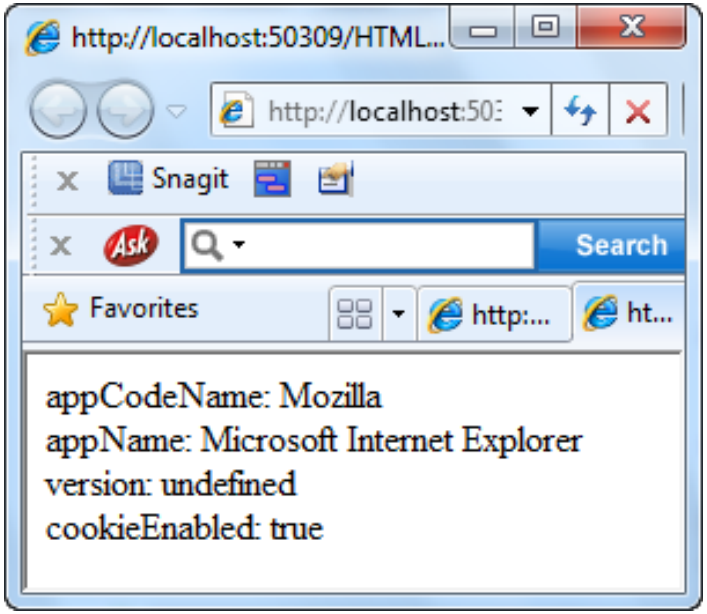
Phương thức

Phương thức	Giải thích
javaEnabled()	Xác định xem trình duyệt có kích hoạt Java hay không

DEMO SỬ DỤNG NAVIGATOR

```
document.write("appCodeName: " + navigator.appCodeName + "<br>");
document.write("appName: " + navigator.appName +"<br>");
document.write("version: " + navigator.version + "<br>");
document.write("cookieEnabled: " + navigator.cookieEnabled);
```





Đối tượng Location

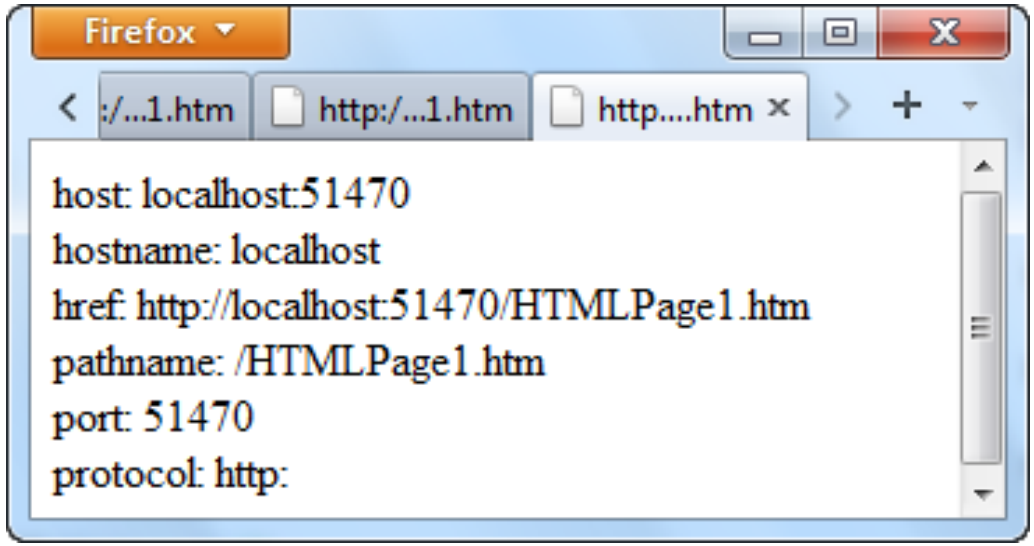
- Quản lý thông tin về URL hiện tại

Thuộc tính	Giải thích
host	Trả về tên **host** và **cổng** của URL
hostname	Trả về tên **host**
href	Trả về toàn bộ **URL**
pathname	Trả về tên **đường dẫn** của URL
port	Trả về **cổng** mà server sử dụng cho URL
protocol	Trả về **protocol** của URL

Phương thức	Giải thích
assign()	**Load** document mới
reload()	**Load lại** document hiện tại

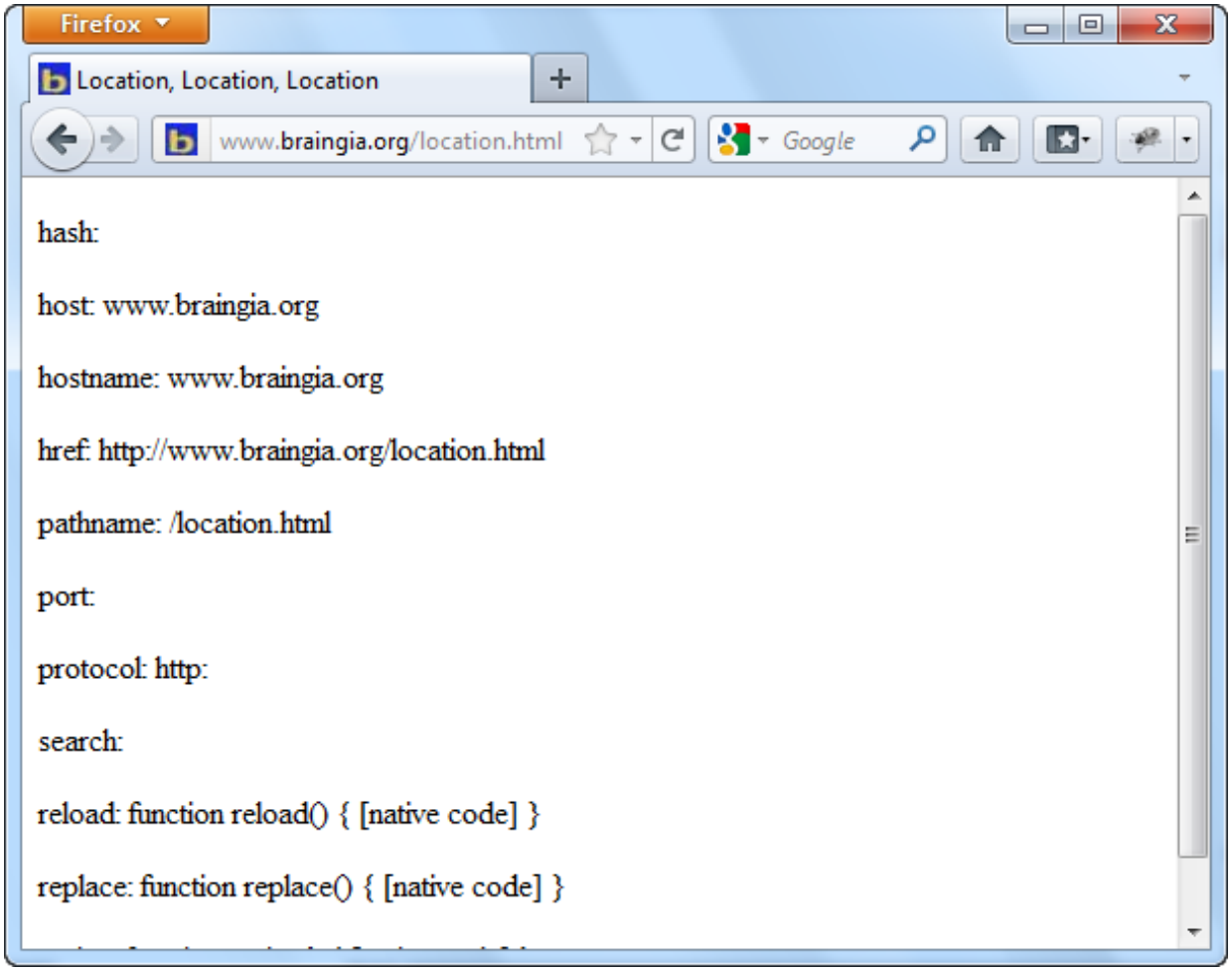
DEMO HIỂN THỊ THÔNG TIN CỦA URL

```
document.write("host: " + location.host + "<br>");
document.write("hostname: " + location.hostname + "<br>");
document.write("href: " + location.href + "<br>");
document.write("pathname: " + location.pathname + "<br>");
document.write("port: " + location.port + "<br>");
document.write("protocol: " + location.protocol + "<br>");
```



DEMO HIỂN THỊ THÔNG TIN CỦA URL

- Vào trang http://www.braingia.org/location.html



DEMO SỬ DỤNG LOCATION ĐỂ MỞ URL MỚI

```
<html >
<head>
<script type="text/javascript">
    function newDoc() {
        window.location.assign("http://www.w3schools.com")
    }
</script>
</head>
<body>
<input type="button" value="Load new document"
onclick="newDoc()" />
</body>
</html>
```

Đối tượng History

- Chứa thông tin về các URL được người dùng truy cập

Thuộc tính	Giải thích
length	Trả về số lượng URL trong danh sách History

Phương thức	Giải thích
back()	Load URL **trước** đó trong danh sách History
forward()	Load URL **sau** đó trong danh sách History
go()	Load URL cụ thể từ History

DEMO SỬ DỤNG HISTORY

- Định nghĩa hàm trong thẻ JavaScript

```
function goBack() {
    history.back();
}

function goNext() {
    history.forward();
}
```

- Gọi hàm

```
<p><a href = "#" onclick="goBack()">Back</a></p>
<p><a href = "#" onclick="goNext()">Next</a></p>
```

TỔNG KẾT BÀI HỌC

- Có rất nhiều phương thức lập trình. Mỗi phương thức phù hợp cho một mục đích riêng. Phương thức lập trình hướng đối tượng được phát triển rộng rãi nhất
- Mỗi đối tượng có các thuộc tính và phương thức riêng
- Các đối tượng có các thuộc tính và phương thức giống nhau thuộc cùng một lớp
- Browser Object Module là tập hợp các đối tượng được xây dựng sẵn giúp lập trình viên thao tác với trình duyệt
- Mỗi trình duyệt hỗ trợ mô hình BOM theo các cách khác nhau nên lập trình viên cần phải tìm hiểu sâu về trình duyệt để viết mã chạy trên nhiều trình duyệt

TỔNG KẾT BÀI HỌC

- Trình duyệt được biểu diễn bằng đối tượng window.
- Đối tượng window có các đối tượng con là document, frames, history, location, navigator, screen
- Đối tượng document đại diện cho nội dung trang web
- Đối tượng history chứa thông tin về các url được người dùng truy cập
- Đối tượng location chứa thông tin về url hiện tại
- Đối tượng navigator chứa thông tin về trình duyệt
- Đối tượng screen chứa thông tin về màn hình

FPT POLYTECHNIC