

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

BÀI 18:

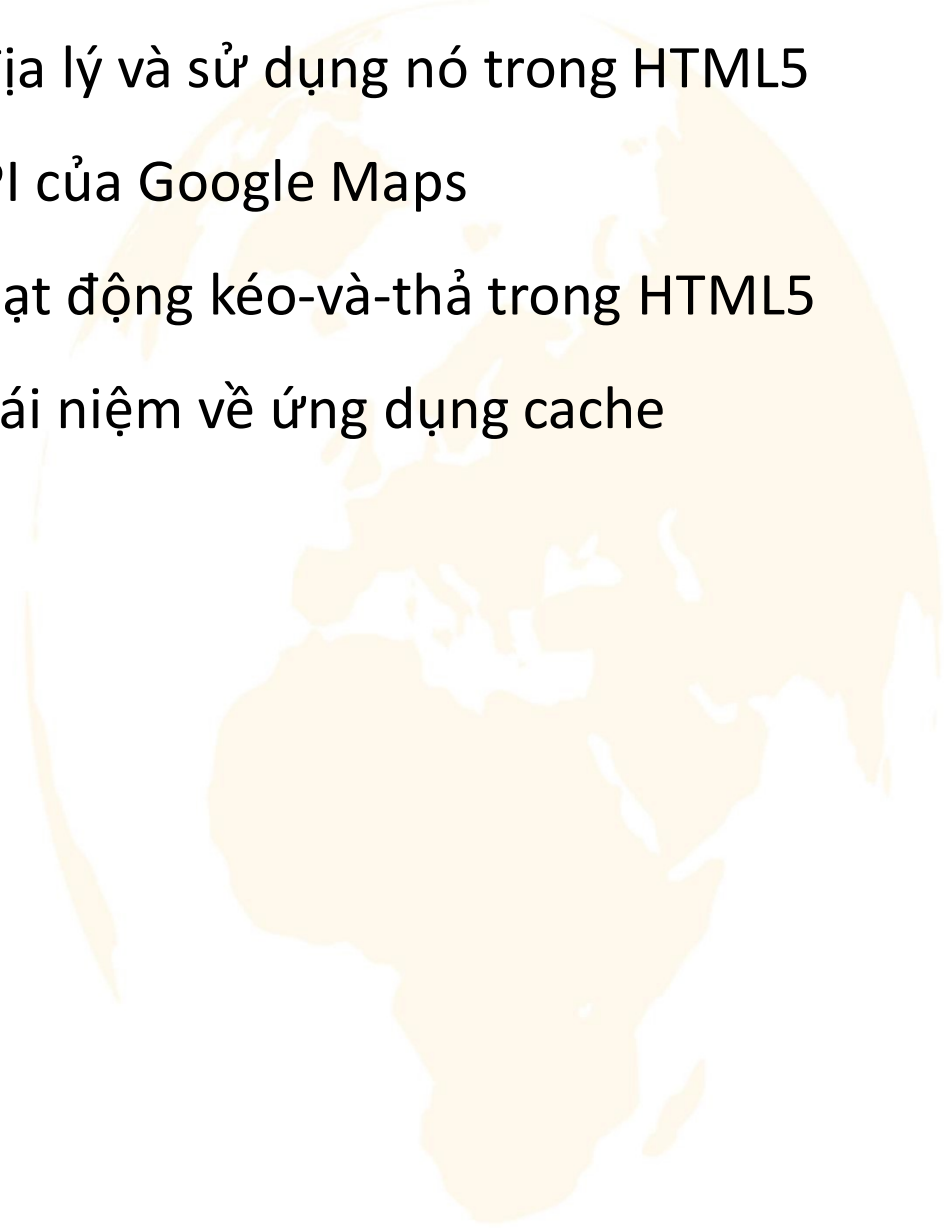
ĐỊNH VỊ ĐỊA LÝ

VÀ

CÁC APIs TRONG HTML5

Mục tiêu

- Giải thích vị trí địa lý và sử dụng nó trong HTML5
- Giải thích các API của Google Maps
- Giải thích các hoạt động kéo-và-thả trong HTML5
- Giải thích các khái niệm về ứng dụng cache

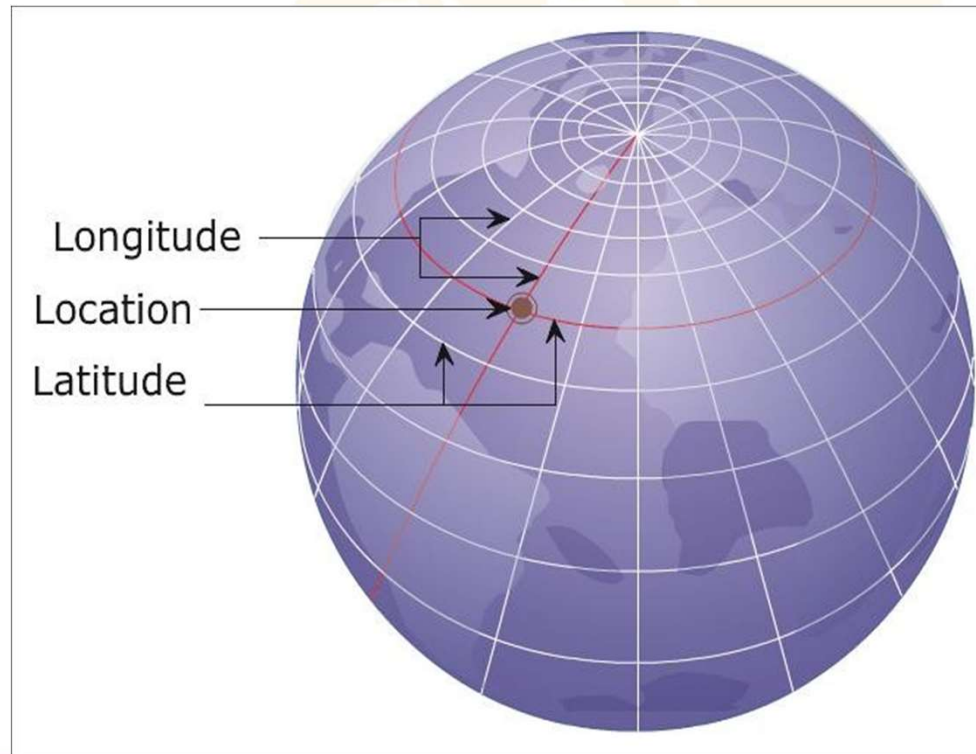


Định vị địa lý 1-2

Định vị trong tính toán thuật ngữ xác định vị trí hiện tại của người dùng trên các thiết bị.

Vị trí của người sử dụng được biểu diễn như là một điểm duy nhất mà bao gồm hai thành phần: vĩ độ và kinh độ.

- Hình dưới đây cho thấy các đại diện của các vĩ độ và kinh độ đối với một vị trí trên toàn thế giới.



Định vị địa lý 2-2

- Các nguồn khác nhau mà qua đó các thiết bị có thể xác định các thông tin về vị trí như sau:

Global Positioning System (GPS)

- GPS là một hệ thống định vị vệ tinh cung cấp thông tin về vị trí trên bất kỳ phần nào trên thế giới.
- Hệ thống GPS được duy trì bởi chính phủ Hoa Kỳ.

IP Address

- Thông tin vị trí có thể được bắt nguồn từ địa chỉ IP được gán cho các thiết bị, như máy tính để bàn, máy in, và như vậy kết nối vào một mạng.

GSM/CDMA Cell IDs

- Chúng được sử dụng bởi các điện thoại di động.

WiFi and Bluetooth MAC address

- Chúng được sử dụng bởi các thiết bị có kết nối mạng không dây.

User Input

- Nó là một công cụ phần mềm có thể được sử dụng trên bất kỳ thiết bị yêu cầu cung cấp thông tin địa điểm.
- Các thông tin lấy được bằng công cụ dựa trên các dữ liệu được cung cấp bởi người sử dụng. Ví dụ, mã vùng.

Định vị địa lý API 1-2

Trong HTML5, Geolocation API là một đặc điểm kỹ thuật của W3C cho việc cung cấp một cách phù hợp để phát triển các ứng dụng web nhận biết vị trí.

Định vị API cung cấp một giao diện cấp cao để lấy thông tin vị trí liên quan đến các thiết bị lưu trữ.

Giao diện ẩn các chi tiết, chẳng hạn như làm thế nào thông tin được thu thập hoặc có phương pháp được sử dụng để lấy thông tin.

Các đối tượng chứa thực hiện Geolocation API là đối tượng Định vị.

Đối tượng này được sử dụng trong JavaScript để lấy thông tin địa lý về các thiết bị lập trình.

Trình duyệt xử lý kịch bản và trả về vị trí với Geolocation API.

Định vị API được hỗ trợ trên hầu hết các trình duyệt hiện đại có sẵn trên máy tính để bàn và điện thoại di động.

Định vị địa lý API 2-2

- Bảng sau liệt kê các trình duyệt hỗ trợ cho Định vị API.

Trình duyệt	Phiên bản hỗ trợ
Safari	5.0+
Chrome	5.0+
Firefox	3.5+
Internet Explorer	9.0+
Opera	10.6+
iOS (Mobie Safari)	3.2+
Android	2.0+
Blackberry	6+

Thực thi đối tượng

Geolocation 1-3

Đối tượng Geolocation có sẵn như là một thuộc tính của đối tượng navigator.

Các đối tượng điều hướng là một đối tượng trình duyệt cho phép người dùng để lấy thông tin về vị trí cụ thể.

Cú pháp:

```
var geolocation = window.navigator.geolocation;
```

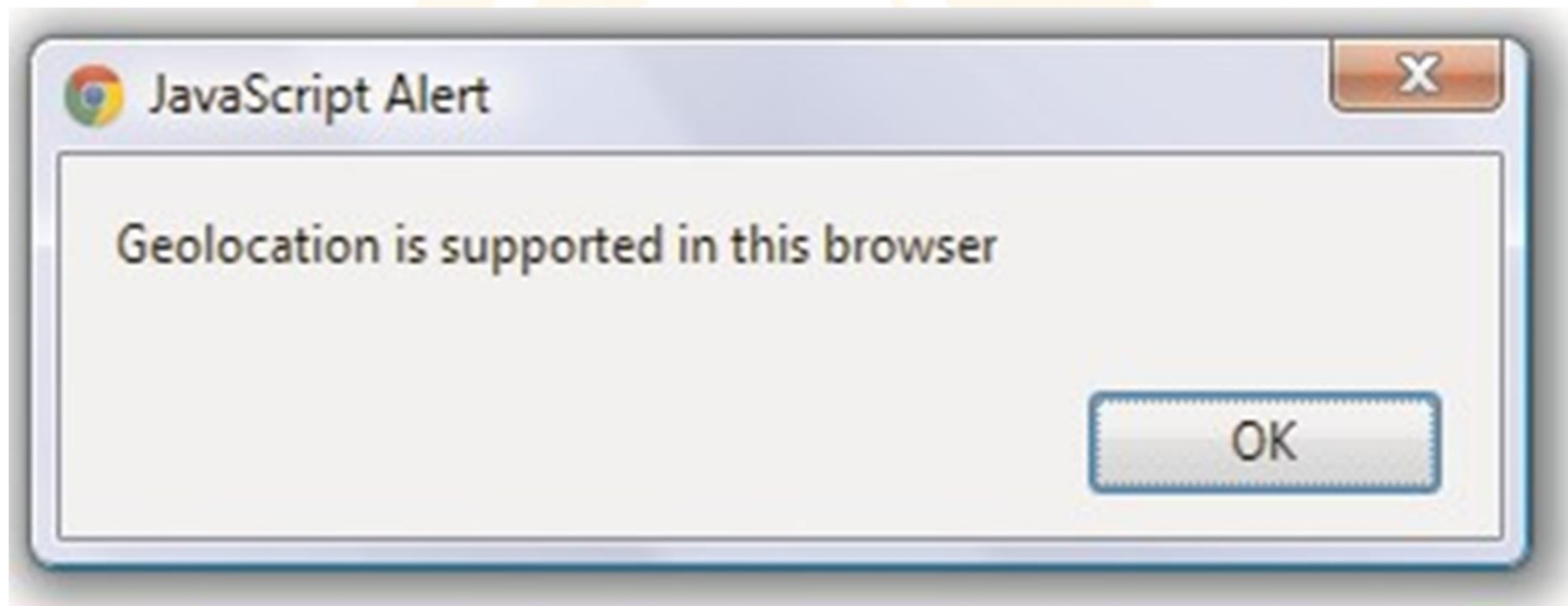
Thực thi đối tượng Geolocation 2-3

- Ví dụ

```
<!DOCTYPE html>
<html>
  <head>
    <title> Testing Support for Geolocation in Browsers</title>
    <script>
      function display_location_enabled()
      {
        // Default message
        var str = "Geolocation is not supported in this browser";
        if (window.navigator.geolocation)
        {
          str = "Geolocation is supported in this browser";
        }
        alert (str);
      }
    </script>
  </head>
  <body>
    <input type="button" value="Geolocation Support"
      onClick="display_location_enabled()"></input>
  </body></html>
```


Thực thi đối tượng Geolocation 3-3

- Kết quả



Các phương thức của Geolocation

- Đối tượng vị trí địa lý cung cấp ba phương pháp có thể được sử dụng để xác định vị trí hiện tại của người dùng.
- Bảng sau liệt kê các phương pháp của đối tượng vị trí địa lý.

Phương thức	Mô tả
<code>getCurrentPosition()</code>	Truy xuất thông tin vị trí địa lý hiện tại của người sử dụng
<code>watchPosition()</code>	Truy xuất thông tin địa lý của các thiết bị điều đặn
<code>clearWatch()</code>	Chấm dứt quá trình xem hiện tại

Lấy thông tin tài khoản 1-5

Vị trí hiện tại của người sử dụng được lấy ra sử dụng phương thức `getCurrentPosition(successCallback, errorCallback, options)`.

Chức năng này chấp nhận ba tham số, trong đó hai là tùy chọn, `errorCallback` và các `Option`.

Tham số đầu tiên, `successCallback` là tên của các chức năng đó được gọi sau khi vị trí của một thiết bị được tìm thấy thành công.

Tham số thứ hai, `errorCallback` là tên của các chức năng sẽ được gọi là, nếu một lỗi xảy ra trong lấy vị trí này.

Tham số cuối cùng, lựa chọn đại diện cho một đối tượng `PositionOptions`.


Lấy thông tin tài khoản 2-5

- Ví dụ

```
<!DOCTYPE html>
<html >
  <head>
<title>Geolocation API</title>
<script>
  function getLocation()
  {
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(showPosition);
    }
    else{
      alert ("Geolocation is not supported in this browser.");
    }
  }
  function showPosition(position)
  {
    alert('Latitude: ' + position.coords.latitude + '\n' +
      'Longitude: ' + position.coords.longitude);
  }
</script>
</head>
```

Lấy thông tin tài khoản 3-5

```
<body>
  <input type="button" value=" Display
    Location"  onClick="getLocation()" />
</body>
</html>
```

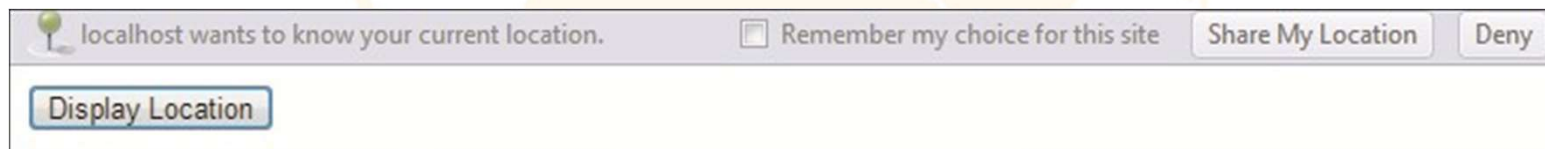


Lấy thông tin tài khoản 4-5

- Bảng sau liệt kê các thuộc tính của đối tượng vị trí.

Thuộc tính	Mô tả
coords	Một đối tượng của kiểu các tọa độ cung cấp đặc tính khác nhau, chẳng hạn như vĩ độ, kinh độ, độ cao, độ chính xác, tốc độ, và như vậy.
timestamp	Một đối tượng kiểu DOMTimeStamp.

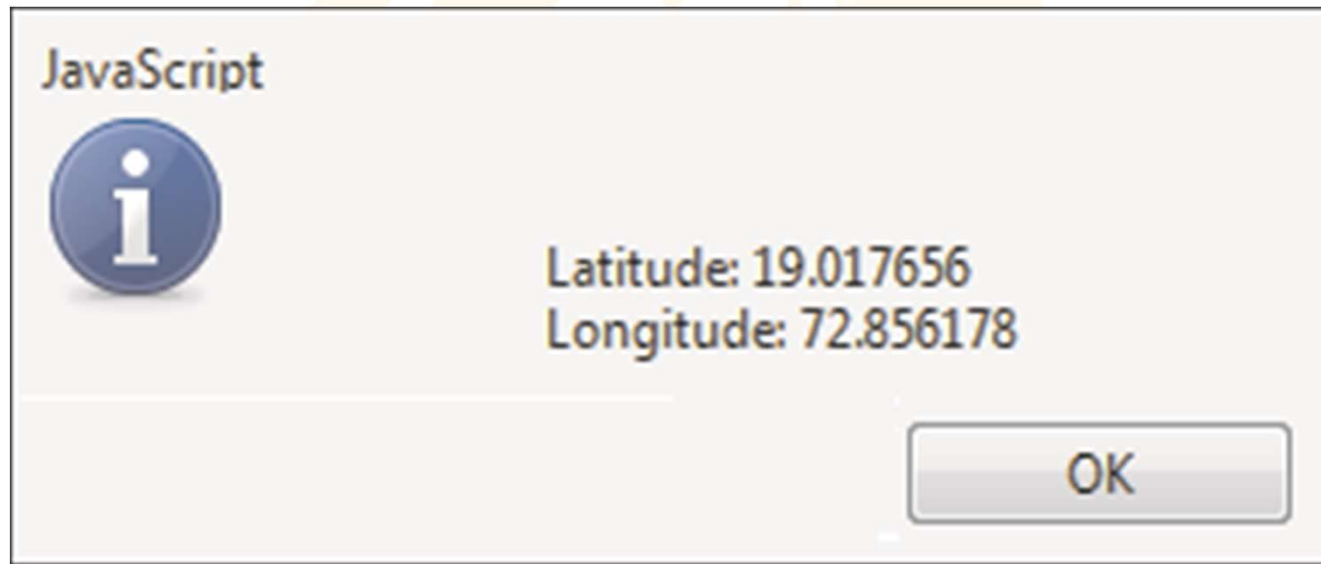
- Hình sau sẽ hiển thị thông báo khi trang web chứa định vị địa lý.



- Trình duyệt tìm kiếm sự cho phép của người sử dụng để chia sẻ thông tin vị trí của họ với ứng dụng.

Lấy thông tin tài khoản 5-5

- Hình dưới đây cho thấy một thông điệp hiển thị vị trí hiện tại của người sử dụng, khi nút Chia sẻ địa điểm được nhấp.



Điều khiển lỗi 1-4

- Một ứng dụng có thể thất bại trong việc thu thập thông tin vị trí địa lý. Trong trường hợp đó, các đối tượng định vị gọi là một hàm errorCallback ().
- Các hàm errorCallback () xử lý lỗi bằng cách lấy một đối tượng PositionError từ API.

➤ HTML

- Đối tượng PositionError chứa thông tin liên quan đến các lỗi xảy ra trong khi tìm kiếm các vị trí địa lý của người dùng.
- Bảng sau liệt kê các thuộc tính của đối tượng PositionError.

Thuộc tính	Mô tả
code	Trả về giá trị số cho các loại lỗi xảy ra.
message	Trả về một thông điệp mô tả chi tiết các lỗi gặp phải. Tin nhắn có thể được sử dụng để gỡ lỗi.

Điều khiển lỗi 2-4

- Bảng sau liệt kê các mã lỗi khác nhau được trả về bởi các mã thuộc tính của đối tượng PositionError.

Mã	Hằng số	Mô tả
1	PERMISSION_DENIED	Ứng dụng không có quyền truy cập Geolocation API.
2	POSITION_UNAVAILABLE	Vị trí của các thiết bị không có thể đạt được.
3	TIMEOUT	Không thể lấy thông tin vị trí trong khoảng thời gian quy định.

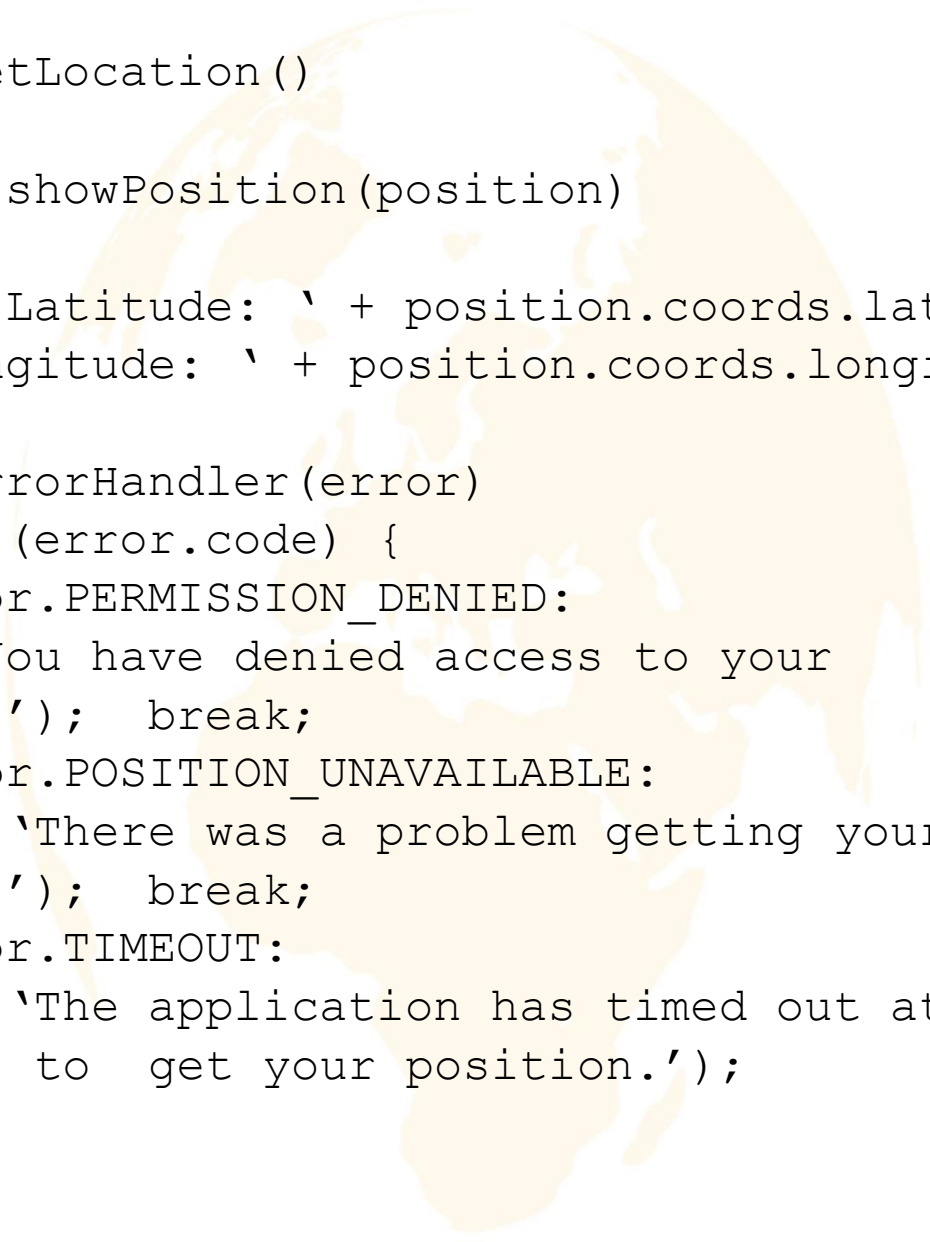
- Mã số câu liên quan chứng minh các thói quen xử lý lỗi cho mã định vị.

```
<!DOCTYPE html>
<html>
<head>

<title>Handling Error</title>
```

Điều khiển lỗi 3-4

```
<script>
  function getLocation()
  {
    function showPosition(position)
    {
      alert('Latitude: ' + position.coords.latitude + '\n'
        + 'Longitude: ' + position.coords.longitude);
    }
    function errorHandler(error)
    {
      switch (error.code) {
        case error.PERMISSION_DENIED:
          alert ('You have denied access to your
            position.');
```



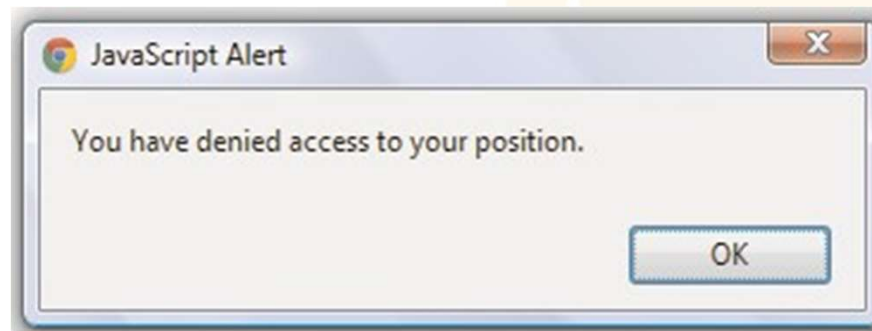
```
          break;
        case error.POSITION_UNAVAILABLE:
          alert ('There was a problem getting your
            position.');
```

```
          break;
        case error.TIMEOUT:
          alert ('The application has timed out attempting
            to get your position.');
```

```
          break;
      }
    }
  }
</script> </head>
```

Điều khiển lỗi 4-4

```
<body>  
  <input type="button" value="Display  
    Location" onClick="getLocation()"/>  
</body> </html>
```



Đối tượng PositionOptions 1-3

- đối tượng PositionOptions là một tham số tùy chọn thứ ba thông qua phương thức `getCurrentPosition()`.
- Đối tượng này xác định thuộc tính đó là tùy chọn và được sử dụng bởi một ứng dụng trong khi lấy các thông tin định vị.
- Bảng sau liệt kê các thuộc tính của đối tượng PositionOptions.

Thuộc tính	Mô tả
<code>enableHighAccuracy</code>	Chỉ ra rằng ứng dụng muốn nhận được kết quả chính xác nhất cho định vị. Giá trị mặc định của thuộc tính là <code>false</code> .
<code>maximumAge</code>	Lấy đối tượng vị trí lưu trữ có thời gian là nhỏ hơn giới hạn quy định <code>maximumAge</code> (trong mili giây).
<code>timeout</code>	Chỉ ra độ dài thời gian tối đa (trong mili giây) mà ứng dụng có thể chờ đợi để có được vị trí các đối tượng.

Đối tượng PositionOptions 2-3

- Ví dụ

```
<script>
    var options = {
        enableHighAccuracy:
            true, maximumAge:
            50000,
            timeout: 60000
        };
    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition
            , errorHandler, options);
        }
        else{
            alert ("Geolocation is not supported in this
            browser.");
        }
    }
    . . .
</script>
```

Đối tượng PositionOptions 3-3

- Trong đoạn mã trên, một đối tượng tên là options được thiết lập với các thuộc tính.
- Thuộc tính, maximumAge cho phép các ứng dụng để sử dụng một đối tượng vị trí lưu trữ mà không phải là lớn hơn 50 giây.
- Ngoài ra, giới hạn thời gian chờ được thiết lập đến 60 giây cho một ứng dụng, trước khi thông báo lỗi.
- Các tùy chọn được truyền vào là tham số thứ ba của phương thức `getCurrentPosition ()`.

Google Maps API 1-5

Google Maps API được sử dụng để hiển thị các địa điểm trên bản đồ dựa trên các giá trị của tọa độ, vĩ độ và kinh độ của họ.

Google Maps API phải được cấu hình trong JavaScript, trước khi nó có thể được tham chiếu thêm trên trang.

Nó chứa một đối tượng bản đồ được thể hiện và hiển thị trên một trang web.

- Sau cú pháp cho thấy cấu hình của Google Maps API trong JavaScript.

Cú pháp:

```
<script src="http://maps.google.com/maps/api/js?sensor=false">  
</script>
```

Google Maps API 2-5

- Ví dụ

```
<!DOCTYPE html>
<html>
  <head>
    <title> Load and Initialize Google Maps </title>
    <style>
      html { height: 100% }
      body { height: 100%; width: 100%; margin: 10% }
      #map_canvas { height: 50%; width: 50% }
    </style>
  <script
    src="http://maps.google.com/maps/api/js?sensor=false"
  >
</script>
```


Google Maps API 3-5

```
function initialize()
{
    // Loading Google Maps
    var num = new google.maps.LatLng(51.528663,-0.173171);    var
    myOptions = {
        zoom: 16,    center:
        num,
        mapTypeId: google.maps.MapTypeId.HYBRID
    };
    var mymap = new google.maps.Map(document.getElementById("
        map_canvas"), myOptions);
    var marker = new google.maps.Marker({
        position: num,    map:
        mymap,
        title:"Lord's Cricket Ground, London!"
    });
}
</script>
</head>
<body onload="initialize()">
    <div id="map_canvas"></div>
</body>
</html>
```

Google Maps API 4-5

- Bảng sau liệt kê một số thuộc tính myOptions.

Thuộc tính	Mô tả
<code>zoom</code>	Thiết lập độ phân giải ban đầu mà bản đồ được hiển thị. Một giá trị hiển thị thấp hơn 0 đại diện cho một bản đồ đầy đủ của Trái Đất. Tương tự như vậy, một giá trị zoom cao sẽ hiển thị một bản đồ với độ phân giải cao.
<code>center</code>	Trung tâm bản đồ trên một điểm cụ thể bằng cách tạo ra một đối tượng kiểu <code>LatLng</code> nắm giữ các tọa độ địa điểm.
<code>mapTypeId</code>	Đặt một loại bản đồ khởi đầu. Các loại bản đồ hỗ trợ là: <code>ROADMAP</code> cho bình thường, <code>SATELITE</code> cho các tấm nhiếp ảnh, <code>HYBRID</code> cho đường và tên thành phố, và các tính năng <code>TERRAIN</code> nước.

Google Maps API 5-5

- Hình dưới đây hiển thị các đối tượng trên trang web tập trung vào Cricket Ground của Chúa ở London.



Theo dõi vị trí người dùng 1-4

- Định vị các đối tượng được sử dụng bởi các API của Google Maps để hiển thị các thông tin định vị trong các ứng dụng.
- Ví dụ

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Map</title>
  <style>
    html, body {
      width: 100%;
      height: 100%;
      padding: 10%
    }
    #map_canvas {
      height: 50%;
      width: 50%;
    }
  </style>
  <script src="http://maps.google.com/maps/api/js?sensor=false">
  </script>
  <script>
```

Theo dõi vị trí người dùng 2-4

```
// Check support for Geolocation in the browser
if (navigator.geolocation) {
    // Locate position and invoke function
```

```
    navigator.geolocation.getCurrentPosition(displayPosition,
errorFunction);
}
else {
```

```
    alert('Geolocation is not enabled in your browser');
    // Success function
```

```
    function displayPosition(position) {
```

```
        var my_lat = position.coords.latitude;
        var my_lng = position.coords.longitude;
        var div_info =
```

```
document.getElementById('user_location');
```

```
        div_info.innerHTML = '<h1> Latitude is
:'+my_lat+' and Longitude is'+my_lng+'</h1>';
```

```
        // Load Google Maps
```

```
        var latlng = new google.maps.LatLng(my_lat,
my_lng);
```

```
        var myOptions = {
```

```
            zoom: 2, //the initial resolution is set
```

```
at which map is displayed
```


Theo dõi vị trí người dùng 3-4

```
center: latlng, //centers the map
mapTypeId: google.maps.MapTypeId.ROADMAP

//sets the map type
};
// Creates the Map object
var map = new
google.maps.Map(document.getElementById("map_canvas"), myOptions);
// Displays icon on the located position
var marker = new google.maps.Marker({
    position: latlng, map: map, title:"User
location"

});
}
// Error callback function
function errorFunction(pos) {
    alert('Error!');
}

}
</script>
</head>
<body>
    <div id="map_canvas"></div> <div id="user_location"></div>
</body>
</html>
```

Theo dõi vị trí người dùng 4-4

- Lệnh `getCurrentPosition()` và lấy vị trí hiện tại của người dùng.
- Sau đó, nó truyền thông tin đến hàm `displayPosition()`, lấy tọa độ, vĩ độ và kinh độ.
- Tọa độ lấy được đưa vào các thuộc tính của đối tượng có tên `myOptions` và khởi tạo các đối tượng bản đồ.
- Cuối cùng, các đối tượng bản đồ được hiển thị cùng với các thông tin vị trí hiện tại trong các thẻ `<div>`.
- Hình dưới đây hiển thị vị trí hiện tại của người dùng trên Google Maps.



Latitude is :19.017656 and Longitude is 72.856178

Kéo và thả (Drag and Drop)

HTML5 xác định các hoạt động kéo-và-thả được dựa trên các sự kiện. Hiện nay, hoạt động kéo-và-thả được hỗ trợ bởi tất cả các trình duyệt chính.

Cơ chế dựa trên sự kiện cho phép các phần tử được sao chép, sắp xếp lại, hoặc bị xóa trên một trang web.

Các thao tác kéo-và-thả liên quan đến việc sử dụng một thiết bị trỏ như chuột trên một phương tiện hình ảnh.

Để thực hiện các thao tác kéo, một sự kiện mousedown được kích hoạt sau nhiều sự kiện mousemove.

Tương tự như vậy, các hoạt động thả được thực hiện khi người dùng nhả chuột.

Ưu điểm của cơ chế kéo-và-thả là nó đã mang lại các hoạt động kéo-và-thả vào mức độ trình duyệt.

Điều này làm cho các lập trình dễ dàng hơn, do đó loại trừ sự cần thiết của mã JavaScript phức tạp bằng văn bản trong các phiên bản HTML trước đó.

Hành động kéo (Drag)

1. Thiết lập các thuộc tính của một phần tử có thể kéo được kéo.
2. Thiết lập một sự kiện ondragstart trên các phần tử mà các dữ liệu lưu trữ đang được kéo.
3. Lưu trữ dữ liệu vào đối tượng datatransfer.

- Ví dụ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Drag and Drop API</title>
  </head>
  <body>
    <div id="div" style="border: red 2px solid;
      height:125px; width:75px; padding: 10px">
      
    </div>
  </body> </html>
```

Sự kiện kéo (Drag Events)

- Trong các giai đoạn khác nhau của hoạt động kéo-và-thả, một số sự kiện được phát ra.
- Những sự kiện này là sự kiện dựa trên chuột.
- Bảng dưới đây liệt kê các sự kiện khác nhau kích hoạt trong các hoạt động kéo.

Sự kiện	Mô tả
dragstart	Kích hoạt khi một phần tử được bắt đầu được kéo bởi người sử dụng.
drag	Kích hoạt khi một phần tử đang được kéo sử dụng một con chuột.
dragleave	Kích hoạt khi kéo và thả hoạt động được hoàn tất.

Đối tượng DataTransfer 1-3

- Đối tượng datatransfer lưu trữ dữ liệu kéo vào các hoạt động kéo-và-thả.
- Nó cho phép nhận và thiết lập các dữ liệu được kéo.
- Nói cách khác, các đối tượng datatransfer giữ dữ liệu trong quá trình kéo-và-thả hoạt động.
- Các datatransfer đối tượng cho phép để xác định hai loại thông tin.
- Đây là như sau:
 - Kiểu dữ liệu của các phần tử kéo
 - Giá trị của các dữ liệu được lưu trữ.

- Ví dụ

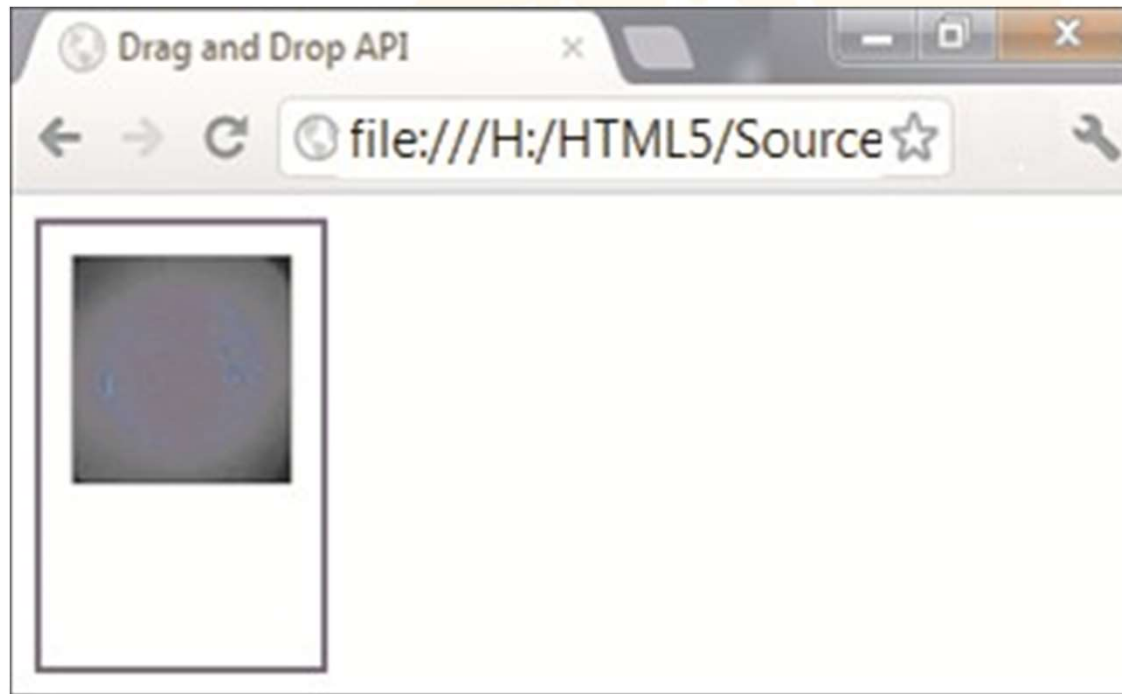
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Drag and Drop API</title>
  <script>
    function drag_image(event)
    {
      event.dataTransfer.setData("image",
event.target.id);
    }
  </script> </head>
```

Đối tượng DataTransfer 2-3

```
<body>
  <div id="div1" style="border: blue 2px solid;
height:125px;
width:75px; padding: 10px">
    
  </div>
</body>
</html>
```

Đối tượng DataTransfer 3-3

- Trong các mã, phần tử đã được thiết lập với một sự kiện dragstart.
- Khi hình ảnh được kéo, sau đó các sự kiện dragstart phát ra và gọi hàm drag_image()?
- Chức năng sử dụng đối tượng datatransfer để lưu trữ các dữ liệu trong quá trình kéo-và-thả hoạt động.
- Chuỗi 'hình ảnh' đại diện cho các kiểu dữ liệu và event.target.id đại diện cho giá trị của thuộc tính id của các phần tử kéo.
- Hình dưới đây cho thấy phần tử hình ảnh sẽ được kéo.



Hành động thả (Drop)

Sau khi các phần tử đã được thiết lập để kéo, nó có thể được thả vào trong một số phần tử trên trang web.

Theo mặc định, các phần tử trên trang không được thiết lập để nhận được các phần tử kéo.

Như vậy, hành vi của các phần tử đóng vai trò là một phần tử thả phải được thay đổi

Điều này có thể được thực hiện bằng cách tạo ra trình nghe sự kiện cho các phần tử thả.

Các phần tử cũng được gọi là phần tử đích.

Sự kiện thả (Drop Events) 1-5

- Đối với bất kỳ phần tử nhận được các hoạt động thả, nó phải gắn với các sự kiện thả.
- Bảng sau liệt kê các sự kiện của hoạt động thả.

Sự kiện	Mô tả
dragenter	Kích hoạt khi một phần tử kéo đang được kéo trên các phần tử mục tiêu lần đầu tiên.
dragleave	Kích hoạt khi một phần tử được kéo bên ngoài phần tử đích.
dragover	Kích hoạt khi một phần tử được kéo bên trong phần tử đích.
drop	Kích hoạt khi một phần tử được giảm trong các phần tử đích

Sự kiện thả (Drop Events) 2-5

- Ví dụ

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Drag and Drop API</title>
  <script>
    function drag_image(event)
    {
      event.dataTransfer.setData("image",
event.target.id);
    }
    function allow_drop(event)
    {
      event.preventDefault();
    }
    function drop_image(event)
    {
      var data=event.dataTransfer.getData("image");
event.target.appendChild(document.getElementById(data));
    }
  </script>
</head>
<body>
  <div id="image1" style="border: 1px solid black; width: 100px; height: 100px; background-color: #f0f0f0; margin-bottom: 10px;">
```


Sự kiện thả (Drop Events) 3-5

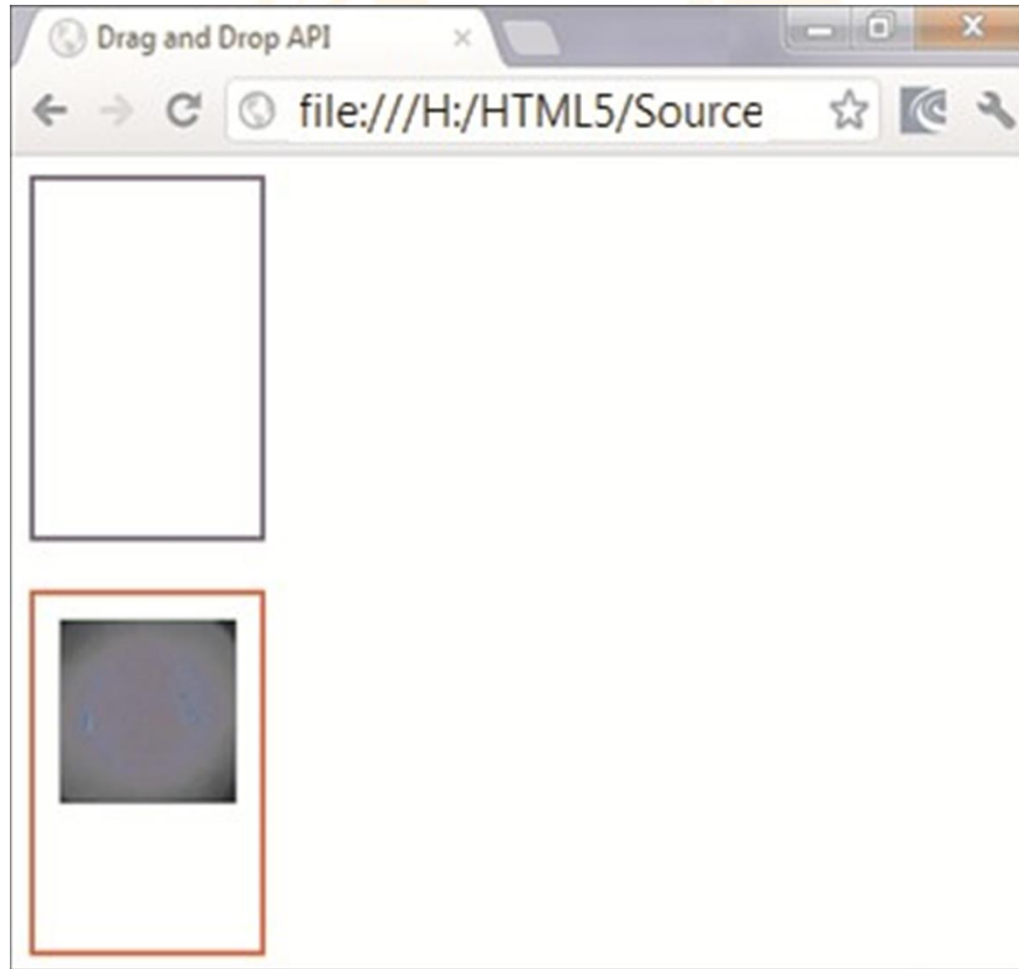
```
</script>
</head>
<body>
  <div id="div1" style="border: blue 2px solid; height:125px;
width:75px; padding: 10px">
    
  </div>
  <br/>
  <div id="div2" style="border: red 2px solid; height:125px;
width:75px; padding: 10px" ondrop="drop_image(event)"
  ondragover="allow_drop(event)">
  </div>
</body>
</html>
```

Sự kiện thả (Drop Events) 4-5

- Trong các mã trên, phần tử <div> với thuộc tính id, thiết lập như là 'div2', có liên quan đến hai người nghe sự kiện cụ thể là, ondragover và ondrop.
- Các ondropover gọi chức năng allow_drop () ngăn ngừa hành vi mặc định của phần tử đích.
- Theo mặc định, trình duyệt không hỗ trợ thả một phần tử trên các phần tử khác.
- Để ngăn chặn hành vi mặc định, câu lệnh event.preventDefault () được gọi.
- Sau đó, sự kiện thả được phát ra vào phần tử đích.
- Nó gọi hàm drop_image() trong đó sử dụng phương thức getData () để lấy hình ảnh được thiết lập như là 'image'.
- Cuối cùng, nó gắn thêm các hình ảnh kéo như một phần tử vào phần tử đích div2.

Sự kiện thả (Drop Events) 5-5

- Hình dưới đây cho thấy đầu ra của các hoạt động thả, sau khi hình ảnh được kéo trên phần tử đích.



Offline Web Applications API

- HTML5 hỗ trợ ứng dụng offline cho phép người dùng làm việc với chúng mà không cần online.
- Các ứng dụng Web offline hoạt động bằng cách ghi tất cả các trang web trên hệ thống cục bộ của người dùng.
- Khái niệm này còn được gọi là ứng dụng Cache.
- Cache Đơn cho phép tất cả các nguồn tài nguyên, chẳng hạn như HTML, JavaScript, hình ảnh, và các trang CSS của một ứng dụng web được lưu trữ cục bộ trên hệ thống.
- Sau đây là các bước có thể được thực hiện lưu nguồn tài nguyên bộ nhớ cache cục bộ trên hệ thống.

1. Tạo một file manifest để xác định các nguồn tài nguyên cần được lưu.

2. Tham khảo các tập tin biểu hiện trong mỗi trang web được thiết kế để sử dụng tài nguyên lưu trữ.

Tạo một tệp Manifest 1-2

- File manifest là một file văn bản định nghĩa cache cho các nguồn tài nguyên được sử dụng bởi các trang web.
- Các tập tin sẽ được lưu với phần mở rộng `.manifest`.
- Mã số câu liên quan chứng minh việc tạo ra một file manifest.

CACHE:

```
# Defines resources to be
  cached.  check.js
  styles.css
  images/figure1.jpg
```

FALLBACK:

```
# Defines resources to be used if non-cached resources cannot
be
# downloaded
  Other_images/ figure2.png
```

NETWORK:

```
# Defines resources that will not be cached.
  figure3.png
```

Tạo một tệp Manifest 2-2

- Sau đây là các phần được định nghĩa trong file manifest..

CACHE

- Phần này xác định các nguồn tài nguyên, chẳng hạn như check.js, styles.css, và figure1.png được lưu trữ tại máy khách.

FALLBACK

- Phần này xác định nguồn tài nguyên thay thế được sử dụng, khi nguồn tài nguyên thực tế là không có sẵn.

NETWORK

- Phần này quy định cụ thể các nguồn tài nguyên để được truy cập khi có một kết nối mạng. Tài nguyên trong phần này không được lưu trữ ..

Khai báo một Manifest 1-3

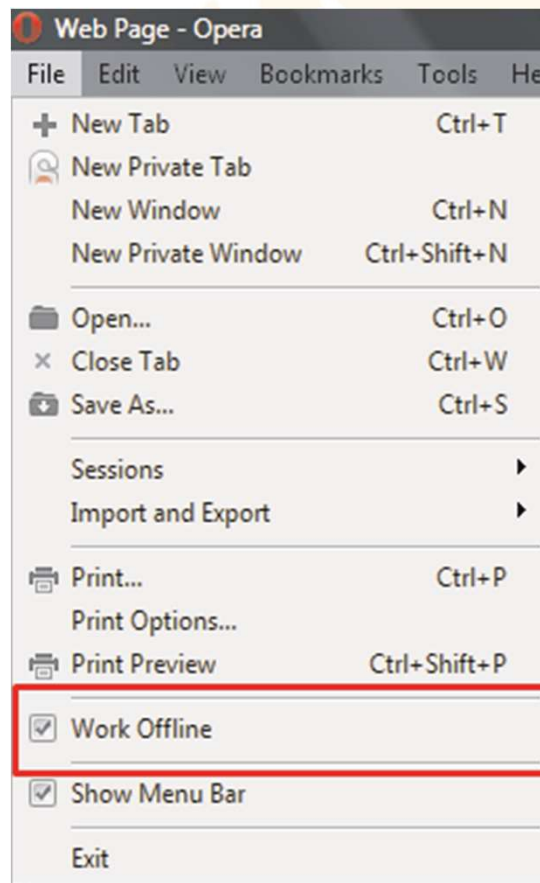
- Để liên kết một manifest với một trang web, chỉ định. File manifest vào thuộc tính có tên `manifest` chỉ ra với các phần tử `html`.
- Mã số câu liên quan chứng minh làm thế nào để thêm tập tin manifest. Trong một tài liệu HTML.

```
<!doctype html>
<html manifest="appcache.manifest">
  <head>
    <title> Web Page </title>
    <link rel="stylesheet" href="styles.css"/>
    <script type="text/javascript" src="check.js"></script>
  </head>
  <body>
    <input type="button" value="click Here..." onClick="display()"/>
    
  </body>
</html>
```

- Trong các mã trên "appcache.manifest" được quy định với thẻ `<html>`.
- Việc giải thích các tập tin manifest tương tự như bất kỳ tài liệu tham khảo tập tin khác.
- Tài liệu này sử dụng một đường dẫn tập tin tương đối, cả hai tập tin biểu hiện và tài liệu HTML được đặt trong cùng một thư mục.
- Theo mặc định, trang web khai báo manifest được lưu trữ tự động.

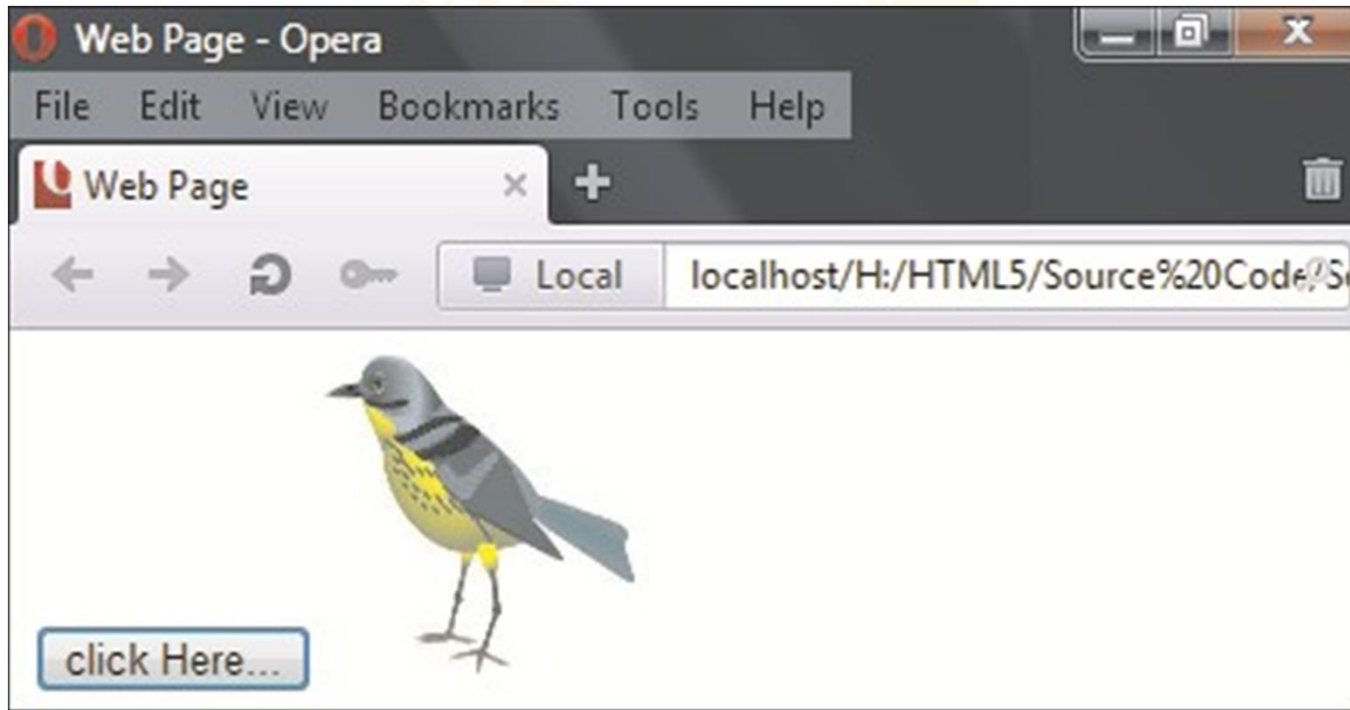
Khai báo một Manifest 2-3

- Lợi ích của việc ứng dụng Cache là nó cải thiện hiệu suất của một trang web bằng cách giảm số lượng yêu cầu thực hiện cho các máy chủ Web.
- Máy chủ Web lưu trữ các ứng dụng web được truy cập trên mạng.
- Con số sau đây cho thấy làm thế nào để kích hoạt chế độ Work Offline trong trình duyệt Opera.
- Điều này cho phép bộ nhớ cache các nguồn tài nguyên của các trang ứng dụng Web cục bộ.



Khai báo một Manifest 3-3

- Hình dưới đây cho thấy các trang web được lưu trữ trong trình duyệt Opera.



- Định vị xác định vị trí hiện tại của người dùng trên các thiết bị.
- Vị trí được biểu diễn như là một điểm duy nhất trên một bản đồ bao gồm hai thành phần: vĩ độ và kinh độ.
- Các Geolocation API là một đặc điểm kỹ thuật được cung cấp bởi W3C cung cấp một cách phù hợp để phát triển các ứng dụng web nhận biết vị trí.
- Google Maps API được sử dụng để hiển thị vị trí của người dùng trên bản đồ.
- Đối tượng của loại bản đồ được tạo ra trong JavaScript, trước khi nó có thể được tham chiếu trong một tài liệu HTML.
- Các hoạt động kéo-và-thả xác định một cơ chế dựa trên sự kiện sử dụng các phần tử trên một trang web có thể được sao chép, sắp xếp lại, hoặc bị xóa.
- HTML5 hỗ trợ ứng dụng offline cho phép người dùng làm việc với họ mà không cần online.

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

THANK FOR WATCH !

