



THỰC HỌC – THỰC NGHIỆP

THIẾT KẾ WEB VỚI HTML5 & CSS3

LÀM VIỆC VỚI HTML5 API

<http://www.poly.edu.vn>

- ⊙ Sử dụng kéo/thả trong HTML
- ⊙ Sử dụng HTML API Geolocation để lấy được vị trí của User
- ⊙ Hiểu được cách lưu trữ dữ liệu cục bộ thông qua HTML web storage



- 📖 Tổng quan HTML5 API
- 📖 Tổng quan HTML Drag/Drop
- 📖 Tổng quan HTML Geolocation
- 📖 Tổng quan HTML Web Storage





PHẦN 1

TỔNG QUAN HTML5 API

- ❑ API (Application Programming Interface) là một giao diện lập trình ứng dụng chứa tập các lệnh và các tiêu chuẩn lập trình để truy cập ứng dụng phần mềm. Với API, bạn có thể thiết kế các sản phẩm được hỗ trợ bởi dịch vụ mà API cung cấp.
- ❑ Trong HTML 5 có một số API mới:
 - ❖ API vẽ 2D – canvas
 - ❖ API giúp lưu trữ dữ liệu giống như cookie.
 - ❖ API để phát video và âm thanh được dùng với các phần tử video và âm thanh mới.
 - ❖ API định vị địa lý
 - ❖ API kéo-và-thả để sử dụng với các thuộc tính draggable



PHẦN 2

TỔNG QUAN VỀ DRAG/DROP

- ❑ **Drag/Drop** (DnD) là một tính năng mạnh mẽ trong HTML5, giúp dễ dàng sao chép, sắp xếp lại và xóa các mục với sự trợ giúp của việc nhấp chuột. Điều này cho phép người dùng có thể dùng chuột nhấc một đối tượng ở vị trí này sang vị trí khác.
- ❑ Trong HTML5 Dnd là một phần của tiêu chuẩn web nên chúng ta có thể kéo mọi đối tượng.
- ❑ Các phiên bản HTML trước HTML5 nếu muốn sử dụng chức năng này, đều phải sử dụng Javascript phức tạp hoặc các thư viện khác như jQuery...
- ❑ HTML5 DnD được hỗ trợ bởi tất cả các trình duyệt hiện đại như Chrome, Firefox...

- ❑ Chúng ta cần thông qua một ví dụ để có thể hiểu được cơ chế hoạt động của drag/drop
- ❑ Đầu tiên ta cần khai báo HTML như sau:

```
<p>Kéo và thả logo vào hình chữ nhật</p>  
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>  
<br>  

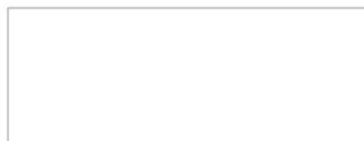
```



```
<script>
  function allowDrop(ev) {
    ev.preventDefault();
  }
  function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
  }

  function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
  }
</script>
```

```
<style>
  #div1 {
    width: 350px;
    height: 70px;
    padding: 10px;
    border: 1px solid #aaaaaa;
  }
</style>
```



Trong ví dụ trên, muốn thực hiện được drag drop. Chúng ta cần quan tâm đến 4 bước:

1. Làm cho một phần tử có thể kéo được

Trước tiên chúng ta cần phải làm cho phần tử có thể kéo được bằng cách thiết lập thuộc tính draggable = "true"

```

```

2. Dữ liệu gì được Kéo đi? – ondragstart and setData()

Trong ví dụ ở trên, thuộc tính ondragstart gọi tới một hàm là drag(event), hàm này quy định dữ liệu gì được kéo đi.

Phương thức dataTransfer.setData() cho phép chúng ta đặt kiểu dữ liệu và giá trị của thẻ được kéo:

```
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}
```

Trong trường hợp này, kiểu dữ liệu là "text" và giá trị là id của thẻ ("drag1").

3. Thả ở đâu? – ondragover

Sự kiện ondragover quy định khu vực nào dữ liệu được phép thả xuống.

Theo mặc định, dữ liệu/thẻ không thể thả trong các thành phần khác của trang. Để cho phép đối tượng được thả xuống, chúng ta sẽ phải thay đổi mặc định của thẻ bằng cách gọi phương thức `event.preventDefault()` cho sự kiện ondragover:

```
ev.preventDefault();
```

4. Thả – ondrop

Khi người dùng thả đối tượng, sự kiện thả sẽ được kích hoạt. Trong ví dụ ở trên, thuộc tính ondrop gọi một hàm là drop(event):

```
function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}
```

Giải thích:

- ❖ Gọi phương thức preventDefault() để thay đổi cách xử lý mặc định của trình duyệt đối với dữ liệu
- ❖ Lấy dữ liệu vừa được thả xuống bằng phương thức dataTransfer.getData(). Phương thức này sẽ trả về kiểu dữ liệu giống như kiểu đã được đặt trong phương thức setData()
- ❖ Ở đây dữ liệu được thả xuống là id của thẻ ("drag1")
- ❖ Chèn thẻ được kéo vào bên trong thẻ được thả



PHẦN 3

TỔNG QUAN VỀ GEOLOCATION

- ☐ Geolocation được sử dụng để xác định vị trí địa lý của người dùng.
- ☐ Để bảo vệ quyền riêng tư, việc xác định vị trí chỉ được thực hiện khi người dùng chấp thuận.
- ☐ Được hỗ trợ bởi hầu hết các trình duyệt hiện đại: chrome,firefox...
- ☐ **Lưu ý:** Việc định vị sẽ cho kết quả chính xác hơn với các thiết bị có GPS, như iPhone, iPad.

- ❑ Để xác định vị trí người dùng bạn sử dụng hàm `getCurrentPosition()`.
- ❑ Dưới đây là ví dụ đơn giản về việc sử dụng Geolocation để lấy tọa độ người dùng:

```
<p>Nhấn vào nút dưới đây để lấy tọa độ của bạn.</p>
<button onclick="getLocation()">Định vị</button>
<p id="demo"></p>
<script>
    var x = document.getElementById("demo");

    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition, showError);
        } else {
            x.innerHTML = "Trình duyệt của bạn không hỗ trợ Geolocation.";
        }
    }

    function showPosition(position) {
        x.innerHTML = "Vĩ độ: " + position.coords.latitude +
            "<br>Kinh độ: " + position.coords.longitude;
    }
</script>
```

Giải thích ví dụ:

- Kiểm tra xem Geolocation có được hỗ trợ hay không
- Nếu được hỗ trợ, chạy hàm `getCurrentPosition()`. Nếu không hỗ trợ, hiển thị thông báo tới người dùng
- Nếu hàm `getCurrentPosition()` được thực hiện thành công, trả về tọa độ của người dùng thông qua hàm `showPosition()`
- Hàm `showPosition()` làm nhiệm vụ hiển thị Kinh độ và Vĩ độ của người dùng
- Trên đây là ví dụ rất căn bản về Geolocation, chưa bao gồm việc xác định và hiển thị các lỗi phát sinh.

- ❑ Tham số thứ 2 của hàm `getCurrentPosition()` được sử dụng để xử lý các lỗi phát sinh trong quá trình định vị. Nó sẽ tự động gọi tới hàm định sẵn nếu xảy ra lỗi:

```
function showError(error) {  
    switch (error.code) {  
        case error.PERMISSION_DENIED:  
            x.innerHTML = "Người dùng từ chối cấp quyền định vị."  
            break;  
        case error.POSITION_UNAVAILABLE:  
            x.innerHTML = "Không có thông tin vị trí."  
            break;  
        case error.TIMEOUT:  
            x.innerHTML = "Hết thời gian gửi yêu cầu định vị."  
            break;  
        case error.UNKNOWN_ERROR:  
            x.innerHTML = "Lỗi chưa xác định."  
            break;  
    }  
}
```


- ❑ Để hiển thị kết quả định vị trên bản đồ, bạn cần tới các dịch vụ cung cấp bản đồ trực tuyến có hỗ trợ Kinh độ và Vĩ độ, chẳng hạn như Google Maps:
- ❑ Trong ví dụ này, chúng ta sử dụng dữ liệu về Kinh độ và Vĩ độ để hiển thị vị trí trên Google maps (bằng ảnh tĩnh).

```
function showPosition(position) {  
    var latlon = position.coords.latitude + "," + position.coords.longitude;  
  
    var img_url = "https://maps.googleapis.com/maps/api/staticmap?center="+latlon+"&zoom=14&size=400x300 &sensor=false&key=YOUR_KEY";  
  
    document.getElementById("mapholder").innerHTML = "<img src='" + img_url + "'>";  
}
```

- Ngoài việc hiển thị vị trí người dùng trên bản đồ, bạn có thể sử dụng Geolocation để:
 - ❖ Hiển thị các địa điểm được yêu thích gần người dùng
 - ❖ Chỉ đường cho người dùng (GPS)
 - ❖ Cập nhật, lưu trữ thông tin về vị trí người dùng

- ❑ Hàm `getCurrentPosition()` sẽ trả về kết quả là một đối tượng (object) nếu định vị thành công.
- ❑ Kinh độ, Vĩ độ và độ là những thuộc tính luôn luôn được trả về. Những thuộc tính khác sẽ được trả về nếu có đủ thông tin cần thiết.

Thuộc tính	Miêu tả
<code>coords.latitude</code>	Vĩ độ
<code>coords.longitude</code>	Kinh độ
<code>coords.accuracy</code>	Độ chính xác của việc định vị
<code>coords.altitude</code>	Độ cao so với mặt nước biển. (m)
<code>coords.altitudeAccuracy</code>	Sự chính xác của độ cao vừa định vị
<code>coords.heading</code>	Hướng người dùng đang đi (độ)
<code>coords.speed</code>	Tốc độ di chuyển (met / giây)
<code>timestamp</code>	Thời điểm phản hồi yêu cầu định vị



PHẦN 4

TỔNG QUAN VỀ WEB STORAGE

- ☐ Với việc sử dụng **local storage** (hay còn gọi là **lưu trữ cục bộ**), ứng dụng web có thể lưu trữ dữ liệu trực tiếp trên trình duyệt của người dùng.
- ☐ Trước khi HTML5 ra đời, dữ liệu web được lưu trữ vào cookie và những dữ liệu này đều được gửi kèm mỗi khi trình duyệt gửi yêu cầu tới máy chủ. Local Storage an toàn hơn cookie và có thể chứa khối lượng dữ liệu lớn (5mb) mà không làm ảnh hưởng tới hiệu suất của website.
- ☐ Tất cả các trang từ cùng một nguồn (cùng tên miền và giao thức), có thể lưu trữ và truy cập vào cùng một khu vực dữ liệu.

- Đồ họa vector có thể mở rộng

- ❑ Local storage cung cấp 2 kiểu lưu trữ dữ liệu trên máy người dùng:
 - ❖ window.localStorage – Lưu trữ dữ liệu không giới hạn thời gian
 - ❖ window.sessionStorage – Chỉ lưu trữ dữ liệu trong một phiên làm việc (dữ liệu sẽ bị xóa khi người dùng đóng cửa sổ trình duyệt)

Trước khi sử dụng local storage, bạn nên kiểm tra trình duyệt của người dùng có hỗ trợ localStorage và sessionStorage hay không:

```
<script>
  if(typeof(Storage) !== "undefined") {
    // Có thể sử dụng localStorage/sessionStorage.
  } else {
    // Trình duyệt không hỗ trợ Local storage.
  }
</script>
```

- Đồ họa vector có thể mở rộng

- ❑ localStorage được sử dụng để chứa những dữ liệu không giới hạn thời gian. Những dữ liệu này sẽ không bị xóa khi người dùng đóng cửa sổ trình duyệt.

```
// Kiểm tra trình duyệt có hỗ trợ local storage không
if (typeof (Storage) !== "undefined") {
    // Lưu trữ
    localStorage.setItem("website", "example.com");
    // Lấy dữ liệu
    document.getElementById("result").innerHTML = localStorage.getItem("website");
} else {
    document.getElementById("result").innerHTML = "Rất tiếc, trình duyệt của bạn không hỗ trợ local storage...";
}
```

- ❑ Cú pháp để xóa "website" trong localStorage như sau:

```
localStorage.removeItem("website");
```

- ☐ **sessionStorage** hoạt động giống như localStorage ngoại trừ việc nó chỉ lưu trữ dữ liệu trên một phiên làm việc. Dữ liệu sẽ bị xóa khi người dùng đóng cửa sổ trình duyệt.
- ☐ Ví dụ sau đây cũng sử dụng để đếm số lần người dùng đã nhấn vào nút, nhưng kết quả chỉ được lưu trong phiên làm việc hiện tại:


```
<p><button onclick="clickCounter()" type="button">Click me!</button></p>
<div id="result"></div>
<p>Click the button to see the counter increase.</p>
<p>Close the browser tab (or window), and try again, and the counter is reset.</p>
<script>
function clickCounter() {
  if (typeof(Storage) !== "undefined") {
    if (sessionStorage.clickcount) {
      sessionStorage.clickcount = Number(sessionStorage.clickcount)+1;
    } else {
      sessionStorage.clickcount = 1;
    }
    document.getElementById("result").innerHTML = "You have clicked the button " +
sessionStorage.clickcount + " time(s) in this session.";
  } else {
    document.getElementById("result").innerHTML = "
Sorry, your browser does not support web storage...";
  }
}
</script>
```

- ☑ Tổng quan HTML5 API
- ☑ Giới thiệu HTML Drag/Drop
- ☑ Giới thiệu HTML Geolocation
- ☑ Giới thiệu HTML Web Storage



thank
you!