

# **PHÁT TRIỂN ỨNG DỤNG WEB**

---

## **BÀI 2.2**

## **HTML5**

# Nội dung

---



- ★ **Giới thiệu HTML5**
- ★ **New Structural Element**
- ★ **Form Element**
- ★ **Graphics (Canvas, SVG, CSS3 2D/3D)**
- ★ **Multimedia (Video Tag, Audio Tag)**
- ★ **Form Attributes**
- ★ **New input type**
- ★ **Web Storage**
- ★ **Hoạt động nhóm**
- ★ **Bài tập**



# Nội dung

---



- ★ **Giới thiệu HTML5**
- ★ **New Structural Element**
- ★ **Form Element**
- ★ **Graphics (Canvas, SVG, CSS3 2D/3D)**
- ★ **Multimedia (Video Tag, Audio Tag)**
- ★ **Form Attributes**
- ★ **New input type**
- ★ **Web Storage**
- ★ **Hoạt động nhóm**
- ★ **Bài tập**



# Giới thiệu HTML5

---



- ◆ HTML là phiên bản thứ năm của HTML cũng là phiên bản mới nhất hiện nay được ra đời từ năm 2009
- ◆ HTML5 bao gồm các mô hình xử lý chi tiết để tăng tính tương thích, mở rộng, cải thiện và chuẩn hóa các tài liệu trên web.
- ◆ Giúp tạo ra các giao diện phức tạp
- ◆ Giúp ta tạo ra các ứng dụng web chạy trên thiết bị di động



# Giới thiệu HTML5 (tt)

---



- ◆ HTML cung cấp các thẻ các đối tượng mới giúp ta hiển thị các đối tượng đa phương tiện một cách đơn giản hiệu quả.
- ◆ HTML5 cung cấp các thẻ để hiển thị các đối tượng đồ họa đơn giản mà hiệu quả
- ◆ Cung cấp các thẻ để tạo ra giao diện một cách đơn giản, dễ nâng cấp và sửa chữa.



# Điểm mạnh HTML5

---



## \* *Khả năng tương thích:*

Giữ nguyên cấu trúc các phiên bản trước. Ra đời nhiều tính năng mới. Phù hợp trên nhiều nền tảng, khả năng sử dụng trên nhiều hệ thống.

## \* *Tính tiện dụng:*

Cú pháp HTML5 khá thoải mái, cú pháp khá đơn giản thiết kế hỗ trợ sẵn bảo mật, và sự tách biệt giữa phần nội dung và trình bày ngày càng thể hiện rõ.

## \* *Khả năng truy xuất rộng rãi:*

HTML5 mang lại sự hỗ trợ tốt hơn cho người dùng, khả năng hỗ trợ API và DOM



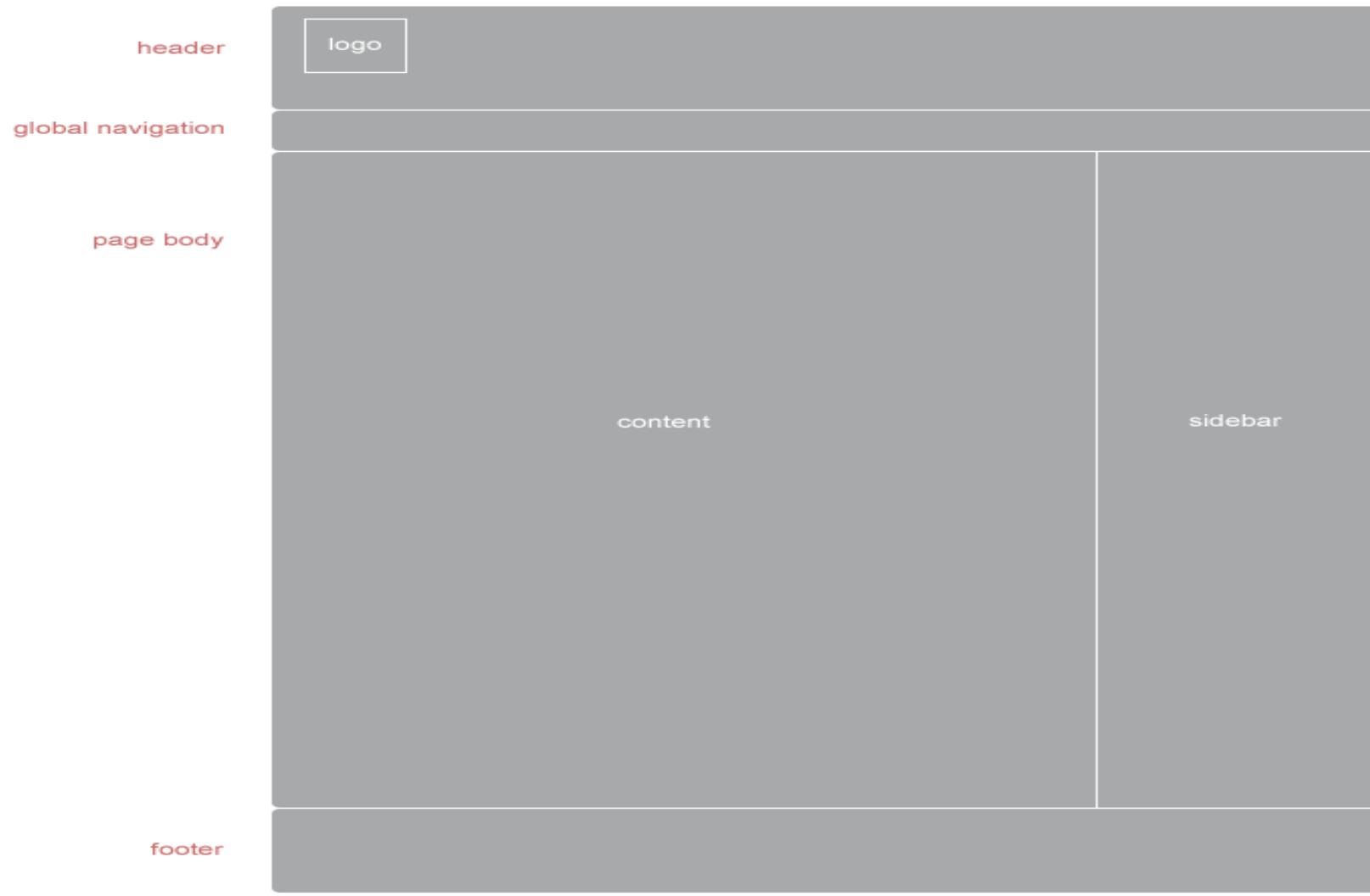
# Nội dung



- \* Giới thiệu HTML5
- \* New Structural Element
- \* Form Element
- \* Graphics (Canvas, SVG, CSS3 2D/3D)
- \* Multimedia (Video Tag, Audio Tag)
- \* Form Attributes
- \* New input type
- \* Web Storage
- \* Hoạt động nhóm
- \* Bài tập



# Cấu trúc trang web



# Cấu trúc trang web (tt)

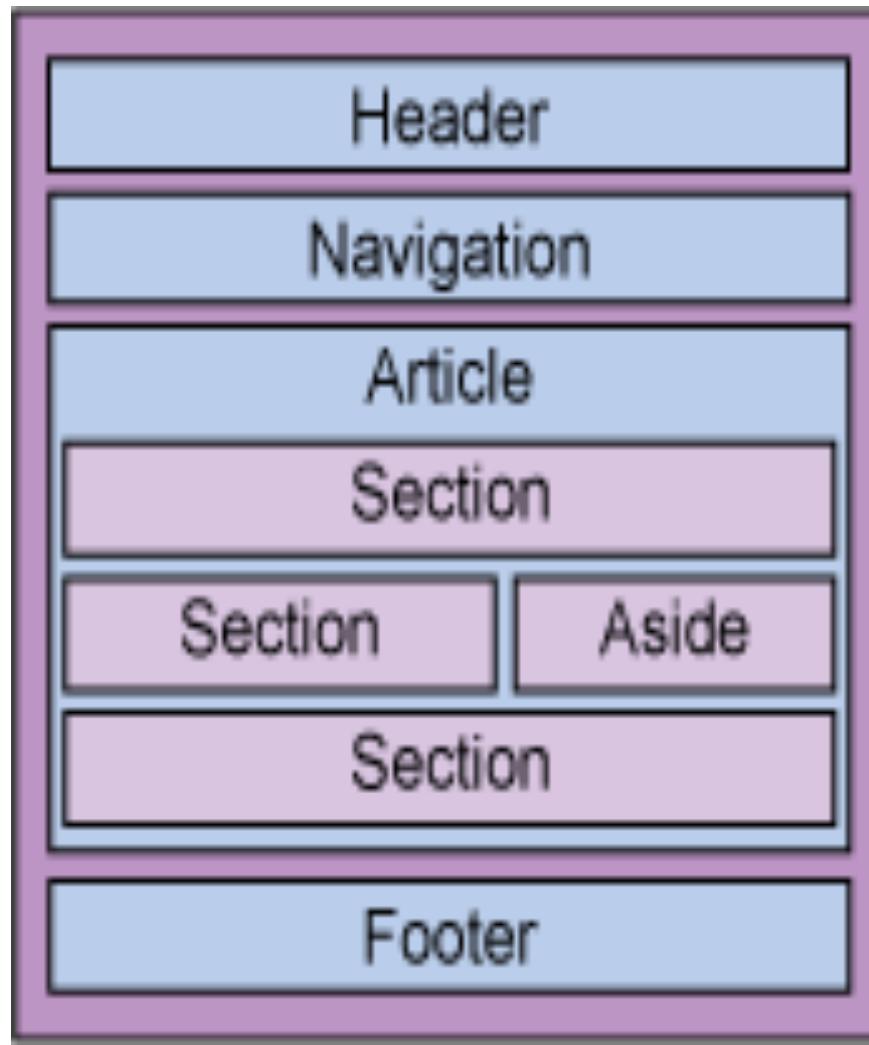


- ◆ **Phần đầu** (header) chứa logo, câu slogan, các liên kết, các banner liên kết, các button, đoạn flash, hoặc các form ngắn như form tìm kiếm,...
- ◆ **Phần liên kết toàn cục**(global navigation) dùng để chứa các liên kết đến những trang quan trọng trong toàn bộ trang, trong phần này có thể chứa thêm các liên kết con (sub navigation).
- ◆ **Phần thân của trang** (page body) phần này chứa phần nội dung chính (content) và phần nội dung phụ (sidebar).
- ◆ **Phần nội dung** (content) phần này chứa nội dung chính cần thể hiện cho người dùng xem.
- ◆ **Phần nội dung phụ** (sidebar) phần này có thể chứa liên kết phụ của từng trang (local navigation), hoặc các banner chứa liên kết liên quan, hoặc có thể dùng để chứa các liên kết quảng cáo,...
- ◆ **Phần cuối trang web** (footer) phần này thường chứa phần liên hệ như: tên công ty, địa chỉ, số điện thoại, mail liên hệ,... và đặc biệt là copyright, hoặc có thể chứa các liên kết toàn trang, các banner liên kết,...



# Các thành phần trong trang html5

---



# Phần Header

---

- ◆ Chứa tiêu đề và phụ đề trang
- ◆ Có thể chứa thông tin mở về một `<section>` và `<article>` ngoài chính trang web đó

```
<header>  
    <h1>Heading Text</h1>  
  
    <p> Text or image can be included here</p>  
  
    <p> Logos are frequently placed here too</p></  
header>
```



# Thẻ hgroup

- ◆ Thẻ `<header>` cũng có thể chứa một thẻ `<hgroup>` dùng để tạo nhóm các tiêu đề với nhau

```
<header>
    <hgroup>
        <h1>Main Heading</h1>
        <h2>Sub-heading </h2>
    </hgroup>
    <p> Text or images can be included here</p>
</header>
```



# Vùng Navigation

- ◆ Dùng để chuyển hướng các trang trong Website
- ◆ Dùng thẻ <nav>

```
<nav>
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Our Products</a></li>
        <li><a href="#">Contact Us</a></li>
    </ul>
</nav>
```

# Vùng Article và Section

---



- ◆ Dùng để hiển thị nội dung chính của trang
- ◆ Thẻ Article và Section dùng để đánh dấu các đoạn
- ◆ Vùng Section: Thể hiện một nội dung cụ thể
- ◆ Vùng Article: thể hiện một phần độc lập của site, chứa nhiều vùng Section và ngược lại



# Ví dụ thẻ Article và Section

```
<article>
```

```
    <section> Content </section>
```

```
    <section> Content </section>
```

```
</article>
```

```
<section>
```

```
    <article> Content </article>
```

```
    <article> Content </article>
```

```
</section>
```

# Vùng Aside

- ◆ chuyên dùng thể hiện những phần trong khung nội dung nhưng được cố định không thay đổi như các menu phụ ở cột right, left.

```
<aside>
```

```
    <h4>Disney in France</h4>
```

```
    <p>Besides Euro Disney, there is a Disneyland in  
California.
```

```
    </p>
```

```
</aside>
```

# Vùng Footer

---

- ◆ Thể hiện nội dung ở phần cuối trang.

```
<footer>
```

```
    <p>Copyright 2011 Acme United. All rights reserved.
```

```
    </p>
```

```
</footer>
```



# Tag <article>



Xác định các phần nội dung cụ thể, độc lập của trang web

## Mã HTML

```
<HTML>
<HEAD><TITLE>The article</TITLE></HEAD>
<BODY>
<article>
    <p> Hoc lap trinh Web </p>
</article>
</BODY>
</HTML>
```

## Kết quả hiển thị

Hoc lap trinh Web

<div class="article">CONTENT</div>



# Tag <nav>

Được dùng để chuyển hướng trang web chính, không dùng để thiết lập các liên kết có chứa trong vùng khác.

## Mã HTML

```
<nav>
  <ul>
    <li><a href="#" title="Lập trình"> Lập Trình </a></li>
    <li><a href="#" title="HTML & CSS"> HTML & CSS </a></li>
    <li><a href="#" title="PHP & MySQL"> PHP & MySQL </a></li>
  </ul>
</nav>
```



# Tag <header>

Định nghĩa phần đầu đề của trang web. Trong HTML4 ta thường dùng <div id=header> để định nghĩa vùng header

## Mã HTML

```
<HTML>
<HEAD><TITLE>The article</TITLE></HEAD>
<BODY>
<header>
    <h1> Hoc lap trinh Web.vn Nơi bắt đầu học làm web </h1>
</header>
</BODY>
</HTML>
```

## Kết quả hiển thị

**Hoc lap trinh Web.vn Nơi bắt đầu học làm web**



# Tag <footer>

Định nghĩa phần (section) dưới cùng của trang web. Thường thì nội dung của thẻ này sẽ chứa các thông tin tác giả, bản quyền, liên hệ...

## Mã HTML

```
<HTML>
<HEAD><TITLE>The article</TITLE></HEAD>
<BODY>
<footer>
    © Bản quyền học bởi web.com
</footer>
</BODY>
</HTML>
```

## Kết quả hiển thị

© Bản quyền học bởi web.com



# Ví dụ



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

Trang chủ    Tin tức    Thông báo    Chương trình học    Diễn đàn

**Tiêu đề bài viết 1**  
Tóm tắt bài viết  
Nội dung bài viết, nội dung bài viết!

- Tác giả

**Tiêu đề bài viết 2**  
Tóm tắt bài viết  
Nội dung bài viết, nội dung bài viết!

- Tác giả

**Tin tức**  
Tin hot trong ngày!

---

Copyright 2012

◆ Demo code của chương trình



# Caption cho hình ảnh



```
<figure>
  
  <figcaption>Hình 1: Câu lạc bộ thể thao</figcaption>
</figure>
```



Hình 1: Câu lạc bộ thể thao



# Nội dung

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Tag <datalist>



Định nghĩa một danh sách tùy chọn, sử dụng <input> xác định giá trị chọn từ danh sách.

## Mã HTML

```
....  
<body>  
<input list="browsers">  
<datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
</datalist>  
<input type="submit" value="Submit"/>  
</body>  
.....
```





# Tag <keygen>

Tạo sự bảo mật cho xác thực người dùng. Nó sẽ định ra một cặp khóa trong form. Khi ta nhấp chọn gửi dữ liệu, hai mã khóa được tạo ra, một khóa private và một khóa public.

## Mã HTML

```
....  
<body>  
<form action="demo_keygen.asp" method="get">  
    Username: <input type="text" name="usr_name">  
    Encryption: <keygen name="security">  
    <input type="submit">  
</form>  
</body>  
.....
```



# *Tag <output>*



Tượng trưng cho một kết quả của một phép tính.

## Mã HTML

```
....  
<body>  
<form oninput="x.value=parseInt(a.value)  
+parseInt(b.value)">  
  <input type="text" name="a"> +<input type="text"  
  name="b" />=  
  <output name="x" for="a b"> </output>  
</form>  
</body>  
.....
```



# Nội dung

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Canvas



- \* Canvas HTML5 là một phần tử vẽ và tạo hình ảnh động, sử dụng JavaScript để vẽ đồ họa trực tiếp trên trang web.
- \* Canvas HTML5 là tag lý tưởng để xây dựng các hình ảnh thú vị, các bản vẽ, các album ảnh, các biểu đồ, các đồ thị, các hình ảnh động, và các ứng dụng bản vẽ nhúng.
- \* Định nghĩa canvas trong tệp HTML:  
*Ví dụ:*

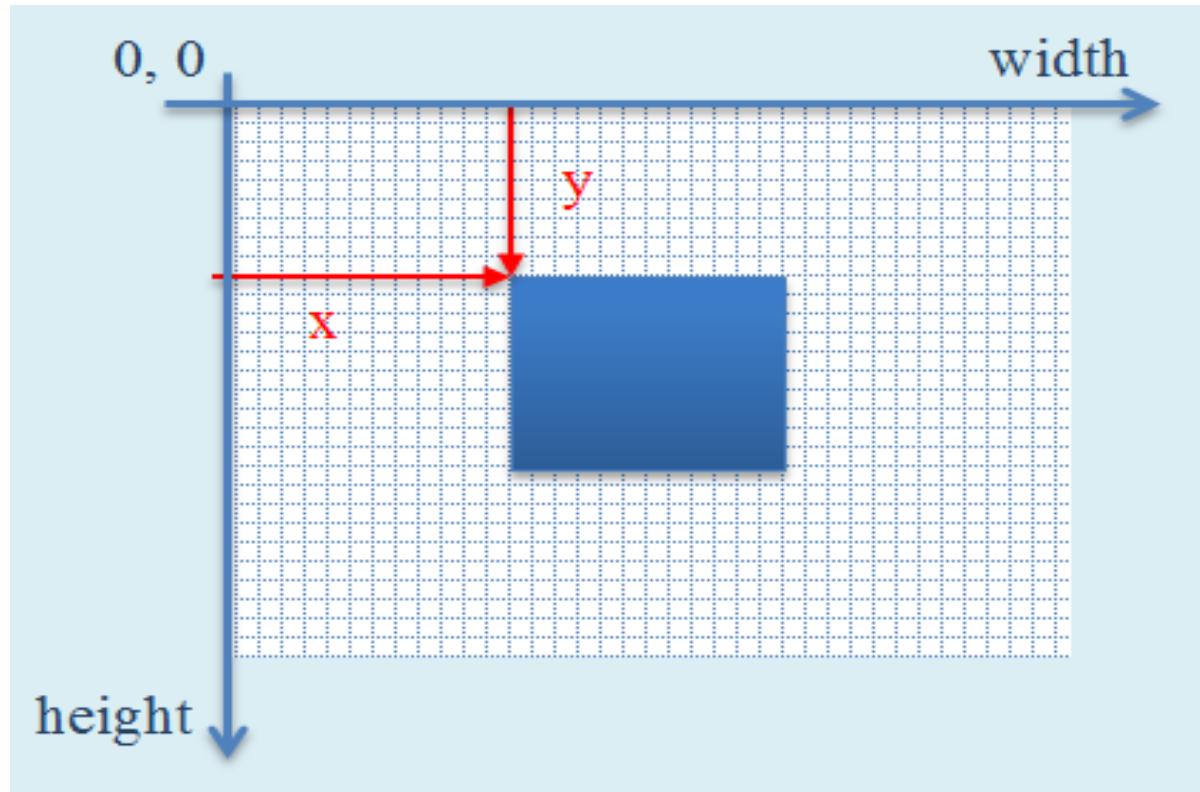
```
<canvas id="myCanvas" width="200" height="200"></canvas>
```

- **Id :** Mã định danh
- **Width :** chiều rộng
- **Height:** chiều dài
- **Class :** Lớp
- **Name:** Tên



# Canvas (TT)

- \* Một canvas có thể được coi như một lưới với số pixel được xác định bởi width và height. Tọa độ trong canvas có gốc nằm ở góc trên trái:



# Canvas (TT)

- ★ JavaScript xác định đối tượng canvas thích hợp, sử dụng phương thức `document.getElementById()`

```
var canvas = document.getElementById("myCanvas");
```

- ★ Mỗi phần tử canvas phải có một định nghĩa ngũ cảnh. Hiện nay, đặc tả chính thức chỉ công nhận môi trường 2D:

```
var context = canvas.getContext("2d");
```

- ★ **Các công cụ vẽ, các hiệu ứng, và các phép biến đổi.**

- **Công cụ vẽ**: các đường kẻ, các hình chữ nhật, các cung tròn, các đường cong Bezier và đường cong bậc hai, các hình tròn và bán nguyệt.
- **Các hiệu ứng**: Tô đầy và các nét, các gradient tuyến tính và xuyên tâm.
- **Các phép biến đổi**: Tỉ lệ, quay tròn, tịnh tiến.



# Canvas (TT)



```
var context = canvas.getContext("2d");
```

## \* Các đường vẽ nét sử dụng phương thức:

- context.beginPath(); **Bắt đầu một đường kẻ mới**
- context.moveTo (x,y); **Nơi bắt đầu đường kẻ mới**
- context.lineTo (x,y); **Xác định điểm cuối cho đường vẽ**
- context.stroke(); **Xác lập vẽ đường kẻ**
- context.lineWidth=x; **Xác định độ rộng của đường kẻ**
- context.strokeStyle=x; **Màu sắc của đường kẻ**
- context.lineCap="x"; **Làm tròn các đầu kẻ**

**Chú ý:** Để vẽ với canvas, chúng ta phải viết trong cặp thẻ  
`<script> </script>`

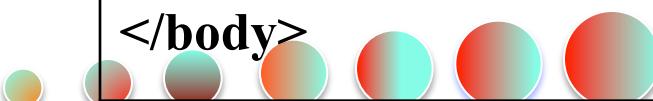


# Canvas (TT)



Mã HTML

```
<script>  
window.onload = function() {  
    var canvas = document.getElementById("myCanvas");  
    var context = canvas.getContext("2d");  
    context.moveTo(0,0)  
    context.lineTo(300,150);  
    context.stroke();  
};</script>  
</head>  
<body>  
<canvas id="myCanvas" width="400" height="200">  
    </canvas>  
</body>
```



# Canvas (TT)



## Mã HTML

```
<script>
    window.onload = function() {
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");
        context.beginPath();
        context.moveTo(50, 50);
        context.lineTo(300,50);
        context.lineWidth = 10;
        context.strokeStyle = "#0000FF";
        context.lineCap = "round";
        context.stroke();
    };
</script>
</head>
<body>
<canvas id="myCanvas" width="400" height="200">
    </canvas>
</body>
```



# Canvas (TT)



```
var context = canvas.getContext("2d");
```

## \* Vẽ hình chữ nhật sử dụng phương thức:

- context.fillRect (x,y,width,height); **Vẽ một hình chữ nhật**
- context.strokeRect (x,y,width,height); **Vẽ nét ngoài của hình chữ nhật.**
- context.clearRect (x,y,width,height); **Làm trong suốt hình chữ nhật.**

### *Chú thích:*

- x, y: **Tọa độ nơi bắt đầu hình chữ nhật sẽ được vẽ.**
- width, height: **chiều dài và chiều rộng tương ứng của hình chữ nhật.**

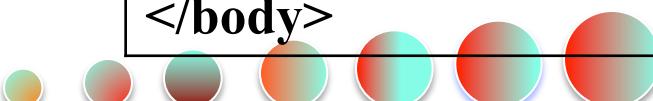


# Canvas (TT)



Mã HTML

```
<script>  
window.onload = function() {  
    var canvas = document.getElementById("myCanvas");  
    var context = canvas.getContext("2d");  
    context.strokeRect(20,20,150,50);  
    context.fillRect(20,80,250,50);  
    context.clearRect(22,82,230,30);  
}; </script>  
</head>  
<body>  
<canvas id="myCanvas" width="400" height="200">  
    </canvas>  
</body>
```



# Canvas (TT)



```
var context = canvas.getContext("2d");
```

## \* Vẽ hình tròn, hình bán nguyệt:

```
context.arc(x, y, r, start, stop, anticlockwise);
```

### *Chú thích:*

- x, y: Tọa độ tâm hình tròn.
  - r : Bán kính hình tròn.
  - start, stop : Điểm đầu và điểm cuối của cung tròn, tính bằng radian.
  - anticlockwise : (Mặc định là False) là một giá trị Boolean. Giá trị True cung được vẽ ngược chiều kim đồng hồ. False cùng chiều kim đồng hồ.
- Sau khi xác định tham số, dùng một phương thức vẽ nét để vẽ.



# Canvas (TT)



## Mã HTML

.....

```
<script>  
window.onload = function() {  
    var canvas = document.getElementById("myCanvas");  
    var context = canvas.getContext("2d");  
    context.arc(300,200,100,0,Math.PI);  
    context.stroke();  
};  
</script>  
</head>  
<body>  
<canvas id="myCanvas" width="400" height="200">  
</canvas>  
</body>
```

# Canvas (TT)



```
var context = canvas.getContext("2d");
```

## \* Vẽ đường cong bậc 2:

`context.moveTo (x,y);`

`context. quadraticCurveTo (controlX, controlY, endX, endY)`

### *Chú thích:*

- x, y: Tọa độ điểm xuất phát.
- controlX, controlY : Tọa độ điểm điều khiển.
- endX, endY : Tọa độ điểm kết thúc.

## \* Vẽ đường cong Bezier.

`context.moveTo (x,y);`

`context. bezierCurveTo (controlX1, controlY1,controlX2,controlY2,`

`endX, endY )`



# Canvas (TT)



Mã HTML

```
<script>  
window.onload = function() {  
    var canvas = document.getElementById("myCanvas");  
    var context = canvas.getContext("2d");  
    context.moveTo(50, 80);  
    context.quadraticCurveTo(75, 25, 100, 80);  
    context.strokeStyle = "red";  
    context.stroke();};</script>  
</head>  
<body>  
<canvas id="myCanvas" width="400" height="200">  
    </canvas>  
</body>
```



# Canvas (TT)



## \* Canvas Gradient

Một *gradien* là một vùng được tô dày và di chuyển từ một màu này sang màu khác, pha trộn các màu ở nơi chúng giao nhau.

**createLinearGradient(x0,y0,x1,y1);** Quét màu xuất phát từ một phía.

**createradialGradient(x0,y0,r0,x1,y1,r1);** Quét màu từ tâm

### Chú thích:

- (x0, y0): (x1, y1): **Tọa độ điểm đầu.Tọa độ điểm cuối.**
- x0, y0 ,r0 : **Tọa độ, bán kính của đường tròn đầu tiên.**
- x1, y1, r1 : **Tọa độ, bán kính của đường tròn thứ hai.**



# Canvas (TT)



## \* SVG Gradient

- Phương thức: **addcolorStop(offset, color)** Xác định màu sẽ được lưu lại ở những giá trị bù cụ thể.

### *Chú thích:*

- offset: **Giá trị bù (0 or 1).**
- Color : **Màu ở giá trị bù.**
- Phương thức: **fillStyle()** Hiển thị gradient.



# Canvas (TT)



Mã HTML

```
<script>  
window.onload = function() {  
    var c=document.getElementById("myCanvas");  
    var ctx=c.getContext("2d"); // Create gradient  
    var grd=ctx.createLinearGradient(0,0,200,0);  
    grd.addColorStop(0,'red');  
    grd.addColorStop(1,'white'); // Fill with gradient  
    ctx.fillStyle=grd;  
    ctx.fillRect(10,10,150,80); };</script>
```

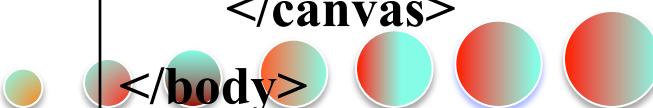
```
</head>
```

```
<body>
```

```
    <canvas id="myCanvas" width="400" height="200">
```

```
        </canvas>
```

```
</body>
```



# SVG (Scalable Vector Graphics)



- \* SVG được sử dụng để xác định các đồ họa vector cho Web
- \* SVG định nghĩa đồ họa ở định dạng XML
- \* SVG đồ họa hình ảnh không ảnh hưởng đến chất lượng khi chúng được phóng to hoặc thay đổi kích thước ảnh

Ưu điểm của SVG:

- **Hình ảnh SVG có thể được tạo ra và chỉnh sửa trên trình soạn thảo văn bản.**
- **Hình ảnh SVG có thể được tìm kiếm, lập chỉ mục, kịch bản , và nén, hình ảnh SVG được mở rộng**



# SVG (Scalable Vector Graphics)

---



- ❑ **Hình ảnh SVG có thể được in với chất lượng cao ở độ phân giải bất kỳ. Có khả năng phóng to mở rộng mà không bị vỡ hình.**
- ❑ **Hình ảnh SVG Zoomable ( và hình ảnh có thể được zoomed mà không làm giảm đi chất lượng màu sắc)**
- ❑ **Các trình duyệt hỗ trợ: Internet Explorer 9+, Firefox, Opera, Chrome, and Safari support inline SVG**



# SVG (Scalable Vector Graphics)

---

## ❑ SVG Shape

**SGV Shape** có nhiều đối tượng : SGV Circle, Rectangle, Ellipse, Line, Polygon, Polyline, Path, Stroking

### *Ví dụ: SVG Circle*

```
<svg>  
<circle cx="100" cy="50" r="40" stroke="black" stroke-width="2"  
fill="red" />  
</svg>
```

### *Ví dụ: SVG Eclipse*

```
<svg>  
  <ellipse cx="300" cy="80" rx="100" ry="50"  
    style="fill:yellow;stroke:purple;stroke-width:5" />  
</svg>
```



# SVG (Scalable Vector Graphics)

---

## ❑ SVG Filter

Dùng để tạo các hiệu ứng mờ cho ảnh

*Ví dụ:*

```
<svg>
<filter id="f1" x="0" y="0">
    <feGaussianBlur in="SourceGraphic" stdDeviation="15" />
</filter>
<ellipse cx="300" cy="80" rx="100" ry="50"
style="fill:yellow;stroke:purple;stroke-width:0;filter:url(#f1)" />
</svg>
```



# SVG (Scalable Vector Graphics)



## ▫ **SVG <feOffset>**

Các yếu tố <feOffset> được sử dụng để tạo ra các hiệu ứng đổ bóng

*Ví dụ:*

```
<svg>
  <defs>
    <filter id="f1" x="0" y="0" width="200%" height="200%">
      <feOffset result="offOut" in="SourceGraphic" dx="20" dy="20" />
      <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10"/>
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow" filter="url(#f1)" />
</svg>
```



# SVG (Scalable Vector Graphics )

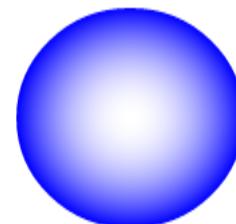
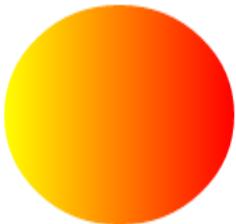
---

## ❑ SVG Gradient

**Gradient là một quá trình chuyển đổi từ một màu sang màu khác.**

## ❑ Có 2 loại SVG Gradients: linear Gradients và radial Gradients

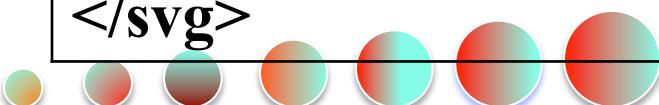
- **SVG <linearGradients>** : Chuyển màu theo độ dốc ngang, dọc hay nghiêng
- **SVG <radialGraients>** : Chuyển màu theo đường dốc xuyên tâm



# SVG (Scalable Vector Graphics)



```
<svg>
<defs>
<linearGradient id="grad1" x1="0%" y1="0%"
x2="100%" y2="0%">
<stop offset="0%" style="stop-
color:rgb(255,255,0);stop-opacity:1" />
<stop offset="100%" style="stop-
color:rgb(255,0,0);stop-opacity:1" />
</linearGradient>
</defs>
<ellipse cx="200" cy="70" rx="85" ry="55"
fill="url(#grad1)" />
</svg>
```



# Drag and Drop

---



- ❑ Cho phép kéo thả một đối tượng từ vị trí này sang vị trí khác
- ❑ Các trình duyệt hỗ trợ: Internet Explorer 9+, Firefox, Opera, Chrome, and Safari support drag and drop.



# NỘI DUNG TRÌNH BÀY

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Multimedia (Tag Video)



- \* Nhúng một video vào web mà không cần hỗ trợ Plugin.

```
<video>
    <source src="E:\Demo.mp4" type="video/mp4">
    <source src="E:\Demo.org" type="video/org">
</video>
```

## **Chú thích:**

- “E:\Demo.mp4” : Đường dẫn đến video.
  - Thẻ `<video>` có thể có nhiều thẻ `<source>`, `<source>` có thể chứa nhiều video khác nhau.
  - Ngày nay có 3 định dạng `<video>` hỗ trợ: mp4, org, webm
- \* Cần được bổ sung các thuộc tính cho Video, để có thể chạy, điều khiển,...được video.



# Multimedia (Tag Video) TT



- \* Thuộc tính kỹ thuật của video.

Thuộc tính	Mô tả
<b>width</b>	<b>Thuộc tính này xác định chiều rộng của vùng hiển thị của video, theo pixel CSS.</b>
<b>loop</b>	<b>Thuộc tính boolean này nếu được chỉ định, sẽ cho phép video tự động lặp trở lại bắt đầu sau khi kết thúc</b>
<b>preload</b>	<b>Thuộc tính này xác định rằng video sẽ được nạp lúc tải trang, và sẵn sàng để chạy. Bỏ qua nếu autoplay được chỉ định.</b>
<b>poster</b>	<b>Đây là một URL của một hình ảnh để hiển thị cho đến khi người sử dụng đóng hoặc chuyển.</b>
<b>src</b>	<b>URL của video để nhúng. Đây là tùy chọn, thay vào đó bạn có thể sử dụng các thẻ &lt;source&gt; trong khối video để xác định video để nhúng</b>



# Multimedia (Tag Video) TT



- ★ Thuộc tính kỹ thuật của video.

Thuộc tính	Mô tả
autoplay	<b>Thuộc tính boolean này nếu được chỉ định, video sẽ tự động bắt đầu.</b>
autobuffer	<b>Thuộc tính boolean này nếu được chỉ định, video sẽ tự động bắt đầu đưa vào bộ đệm ngay cả khi nó không được thiết lập bật tự động.</b>
controls	<b>Cho phép người sử dụng để kiểm soát phát lại video, bao gồm âm lượng, tìm kiếm, và tạm dừng / tiếp tục phát lại.</b>
height	<b>Thuộc tính này xác định chiều cao của vùng hiển thị của video, theo pixel CSS.</b>



# Multimedia (Tag Video) TT

---

```
<!DOCTYPE HTML>
<html>
<body>
<video width="800" height="500" controls="controls"
    autoplay="autoplay">
    <source src="E:\Demo.mp4" type="video/mp4 ">
    <source src="E:\Demo.webm" type="video/webm">
</video>
<body>
</html>
```



# Multimedia (Tag Audio)



- \* Nhúng một nội dung âm thanh vào trang web.

```
<audio>
    <source src="E:\Demo.mp3" type="video/mp3">
    <source src="E:\Demo.ogg" type="video/ogg">
</audio>
```

## **Chú thích:**

- “E:\Demo.mp4” : Đường dẫn đến audio.
  - Thẻ `<audio>` có thể có nhiều thẻ `<source>`, `<source>` có thể chứa nhiều audio khác nhau.
  - Ngày nay có 3 định dạng `<audio>` hỗ trợ: mp3, ogg, wav
- \* Cần được bổ sung các thuộc tính cho audio, để có thể điều khiển,...được audio



# Multimedia (Tag Audio)



- \* Thuộc tính kỹ thuật của audio.

Thuộc tính	Mô tả
autoplay	Thuộc tính boolean này nếu được chỉ định, âm thanh sẽ tự động bắt đầu chạy
autobuffer	Thuộc tính boolean này nếu được chỉ định, âm thanh sẽ tự động bắt đầu đưa vào bộ đệm ngay cả khi nó không được thiết lập bật tự động.
controls	Cho phép người sử dụng để kiểm soát phát lại âm thanh, bao gồm âm lượng, tìm kiếm, và tạm dừng / tiếp tục phát lại.
loop	Thuộc tính này xác định cho phép âm thanh tự trở lại bắt đầu, sau khi kết thúc.



# Multimedia (Tag Audio)



- \* Thuộc tính kỹ thuật của audio.

Thuộc tính	Mô tả
<b>preload</b>	<b>Thuộc tính này xác chỉ ra rằng âm thanh sẽ được nạp vào trang, và sẵn sàng để chạy. Bỏ qua nếu autoplay là hiện tại.</b>
<b>src</b>	<b>Đây là tùy chọn, thay vào đó bạn có thể sử dụng các thẻ &lt;source&gt; trong khối video để xác định video nhúng</b>



# Multimedia (Tag Audio)



## \* Quản lý sự kiện Media.

Thẻ âm thanh và thẻ video HTML5 có thể có một số thuộc tính để điều khiển các chức năng khác nhau của điều khiển bằng cách sử dụng Javascript:

Sự kiện	Mô tả
<b>abort</b>	<b>Sự kiện được tạo ra khi phát lại được hủy.</b>
<b>canplay</b>	<b>Sự kiện được tạo khi đủ dữ liệu có sẵn các đa phương tiện có thể được phát.</b>
<b>error</b>	<b>Sự kiện được tạo ra khi một lỗi xảy ra.</b>
<b>ended</b>	<b>Sự kiện được tạo ra khi phát lại hoàn tất.</b>
<b>loadeddata</b>	<b>Sự kiện này được tạo ra khi frame đầu tiên của đa phương tiện đã tải xong.</b>



# Multimedia (Tag Audio) TT



## \* Quản lý sự kiện Media.

Sự kiện	Mô tả
loadstart	<b>Sự kiện</b> được tạo ra khi việc tải đa phương tiện bắt đầu.
pause	<b>Sự kiện</b> được tạo ra khi phát lại bị tạm dừng.
play	<b>Sự kiện</b> được tạo ra khi phát lại bắt đầu hoặc phục hồi lại..
progress	<b>Sự kiện</b> được tạo ra theo định kỳ để thông báo sự tiến trình của việc tải về đa phương tiện.
ratechange	<b>Sự kiện</b> này được tạo ra khi thay đổi tốc độ phát lại.
seeked	<b>Sự kiện</b> được tạo ra khi một thao tác tìm kiếm hoàn tất.





# Multimedia(Tag Audio) TT

## \* Quản lý sự kiện Media.

Sự kiện	Mô tả
seeking	<b>Được tạo ra khi một hoạt động tìm kiếm bắt đầu.</b>
suspend	<b>Sự kiện này được tạo ra khi tải của đa phương tiện bị đình chỉ.</b>
volumechange	<b>Được tạo ra khi thay đổi âm lượng.</b>
waiting	<b>Sự kiện này được tạo ra khi hoạt động yêu cầu (như phát lại) bị trì hoãn trong khi chờ hoàn thành các hoạt động khác (chẳng hạn như "tìm kiếm").</b>



# Multimedia(Tag Audio) TT

---



```
<!DOCTYPE HTML>
<html>
<body>
<audio controls autoplay loop="loop">
    <source src="E:\Keyshia Cole - Falling Out.mp3"
    type="audio/mp3" />
    <source src="E:\Keyshia Cole - Falling Out.ogg"
    type="audio/ogg" />
    <p> Your browser does not support the <audio> element.</p>
</audio>
<body>
</html>
```



# NỘI DUNG TRÌNH BÀY

---



- \* New Structural Element
- \* Form Element
- \* Graphics (Canvas, SVG, CSS3 2D/3D)
- \* Multimedia (Video Tag, Audio Tag)
- \* Form Attributes
- \* New input type
- \* Web Storage
- \* Hoạt động nhóm
- \* Bài tập



# Attribute of Form(Autocomplete)



- \* Thuộc tính **autocomplete** cho phép trình duyệt sẽ tự động lưu lại các dấu vết các giá trị mà người dùng nhập trên web trước đó.

```
<form action="demo_form.php" autocomplete="on">
    First name:<input type="text" name="fname"><br>
    Last name: <input type="text" name="lname"><br>
    E-mail: <input type="email" name="email"
        autocomplete = "on"><br>
    <input type="submit">
</form>
```



# Attribute of Form (Novalidate)



- ★ Thuộc tính **novalidate** làm cho dữ liệu input không được xác nhận khi gửi đi.
- ★ Thuộc tính **novalidate** là thuộc tính Boolean.
- ★ Hỗ trợ các trình duyệt: Opera, Chrome, Firefox

```
<form action="demo_form.php" novalidate >
    First name:<input type="text"
    name="fname"><br>
    E-mail: <input type="email" name="email"><br>
    <input type="submit">
</form>
```



# Attribute of input (Autofocus)



- ★ Thuộc tính **autofocus** được khai báo. Khi trang web được load, thì con trỏ sẽ nằm ở vị trí input được khai báo.
- ★ Hỗ trợ các trình duyệt :Opear, Chrome, Firefox, Safari

```
First name:<input type="text" name="fname"  
autofocus="autofocus">
```





# Attribute of input (Form)

- \* Thuộc tính **form** được khai báo để cho biết các input thuộc form nào?
- \* Hỗ trợ các trình duyệt :Opearo, Chrome, Firefox, Safari

```
<form action="demo_form.php" id="form1">  
First name: <input type="text"  
name="fname"><br>  
<input type="submit" value="Submit">  
</form> Last name:<input type="text"  
name="lname" form="form1">
```



# Attribute of input (formaction)



- \* Thuộc tính **formaction** giúp chúng ta có thể tùy chọn file nào sẽ xử lý dữ liệu nhập vào. Thuộc tính **formaction** sẽ ghi đè lên thuộc tính **action** của thẻ form
- \* Hỗ trợ các trình duyệt :Opear, Chrome, Firefox, Safari

```
<form action="demo_form.php">  
First name: <input type="text" name="fname"><br>  
Last name: <input type="text" name="lname"><br>  
<input type="submit" value="Submit"><br>  
<input type="submit" formaction="demo_admin.php"  
value="Submit as admin">  
</form>
```



# Attribute of input (formenctype)



- \* Thuộc tính **formaction** giúp chúng mã hóa dữ liệu “**multipart/form-data**” khi gửi đến server. Thuộc tính formenctype sẽ ghi đè lên thuộc tính enctype của phần tử `<form>`.
- \* Hỗ trợ các trình duyệt :Opearo, Chrome, Firefox, Safari

```
<form action="demo_post_enctype.asp"
      method="post">
  First name: <input type="text"
                name="fname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/
                form-data"
                value="Submit as Multipart/form-data">
</form>
```



# Attribute of input (formmethod)



- \* Thuộc tính **formmethod** quy định phương thức gửi form đến phần xử lý trên sever. Thuộc tính formmethod ghi đè lên thuộc tính thuộc tính của phần tử form.
- \* Hỗ trợ các trình duyệt :Opearo, Chrome, Firefox, Safari

```
<form action="demo_form.asp" method="get">
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
<input type="submit" value="Submit">
<input type="submit" formmethod="post"
      formaction="demo_post.php"
      value="Submit using POST">
</form>
```



# Attribute of input(formnovalidate)

- ★ Thuộc tính **formnovalidate** làm cho form không bắt lỗi dữ liệu người dùng nhập vào. Thuộc tính **formnovalidate** sẽ ghi đè lên thuộc tính novalidate của <form>.
- ★ Hỗ trợ các trình duyệt :Opear, Chrome, Firefox.

```
<form action="demo_form.asp">  
E-mail: <input type="email" name="userid"><br>  
<input type="submit" value="Submit"><br>  
<input type="submit"  
      formnovalidate="formnovalidate"  
      value="Submit without validation">
```



# Attribute of input (formtarget)

- ★ Thuộc tính **formnovalidate** được khai báo tùy theo giá trị của target sau khi submit dữ liệu thì sẽ mở trang mới hoặc cửa sổ mới,..  
Thuộc tính formtarget ghi đè thuộc tính target của <form>
- ★ Hỗ trợ các trình duyệt :Opear, Chrome, Firefox, Safari.

```
<form action="demo_form.asp">
    First name: <input type="text"
        name="fname"><br>
    Last name: <input type="text"
        name="lname"><br>
    <input type="submit" value="Submit as normal">
    <input type="submit" formtarget="_blank"
        value="Submit to a new window">
</form>
```

# Attribute of input (list)



- \* Thuộc tính list hiển thị ra một <datalist> chứa các dữ liệu mà ta định sẵn để thuận tiện cho việc nhập liệu.
- \* Hỗ trợ các trình duyệt :Opear, Chrome, Firefox.

```
<datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
</datalist>
<input type ="submit">
```



# Attribute of input (min-max)



- \* Thuộc tính min and max xác định giá trị nhỏ nhất và lớn nhất của thẻ **<input>**.
- \* Hỗ trợ các trình duyệt :Opear, Chrome.

**Nhập thời gian sau ngày 2000-01-01:**

```
<input type="date" name="bday"  
      min="2000-01-02">  
</br>
```

**Số lượng (từ 1 đến 5):**

```
<input type="number" name="quantity" min="1"  
      max="5">  
</br>  
<input type="submit" value="Gửi" />
```



# Attribute of input (pattern,placeholder,required)



- ★ Thuộc tính **pattern** tạo ra sự ràng buộc đối với dữ liệu nhập vào, hỗ trợ các trình duyệt :Opear, Chrome, Firefox, Safari.
- ★ Thuộc tính **placeholder** tạo ra giá trị mẫu trên thẻ **<input>**, giá trị mẫu sẽ bị xóa đi nếu ta chọn đến trường của nó. Hỗ trợ trình duyệt Opera, Chrome, Firefox, Safari.
- ★ Thuộc tính required có 2 giá trị có hoặc không. Thuộc tính này quy định một trường nhập liệu bắt buộc không được rỗng. Hỗ trợ Opera, Chrome, Firefox



# Nội dung

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# New Input Type



HTML5 có một số loại input mới cho các hình thức . Những tính năng mới cho phép kiểm soát đầu vào và xác nhận tốt hơn.

Các loại input mới như: color, date, time, email, url,.....

- ◆ Input dạng color

```
<input type="color" name="" />
```

- ◆ Input dạng date

```
<input type="date" name="" />
```

- ◆ Input dạng datetime

```
<input type="datetime" name="" />
```



# New Input Type

---



- ❖ Input dạng datetime-local

```
<input type="datetime-local" name="" />
```

- ❖ Input dạng email

```
<input type="email" name="" />
```

- ❖ Input dạng month

```
<input type="month" name="" />
```

- ❖ Input dạng number

```
<input type="number" name="" min="3" max="10" step="" />
```

- ❖ Input dạng range

```
<input type="range" name="" min="3" max="10" step="" />
```



# New Input Type

---



- ❖ Input dạng tell

```
<input type="tell" name="" />
```

- ❖ Input dạng time

```
<input type="time" name="" />
```

- ❖ Input dạng url

```
<input type="url" name="" />
```

- ❖ Input dạng week

```
<input type="week" name="" />
```



# Nội dung

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Web Storage



- \* **Web storage** : Cho phép lưu trữ thông tin, dữ liệu bằng trình duyệt web của người dùng.

*Đối tượng của Web storage:*

- **localStorage**: store data không thời hạn
- **sessionStorage**: store data cho một session

**Kiểm tra trình duyệt có hỗ trợ ?**

```
if(typeof(Storage)!=="undefined")
{
    // Yes! localStorage and sessionStorage support!
    // Some code.....
}
else
{
    // Sorry! No web storage support..
}
```





# Web Storage (TT)

- \* Dữ liệu lưu trữ trong Storage chỉ là kiểu chuỗi. Mỗi đối tượng Storage là một danh sách các cặp key/value, đối tượng này bao gồm các thuộc tính và phương thức:

Tên	Mô tả
<b>length</b>	<b>Số lượng cặp key/value có trong đối tượng.</b>
<b>key(n)</b>	<b>Trả về tên của key thứ n trong danh sách.</b>
<b>getItem(key)</b>	<b>Trả về value được gắn với key.</b>
<b>setItem(key,value)</b>	<b>Thêm hoặc gán một cặp key/value vào danh sách.</b>
<b>removeItem(key)</b>	<b>Xóa cặp key/value khỏi danh sách.</b>
<b>clear</b>	<b>Xóa toàn bộ dữ liệu trong danh sách.</b>



# Web Storage(TT)

---

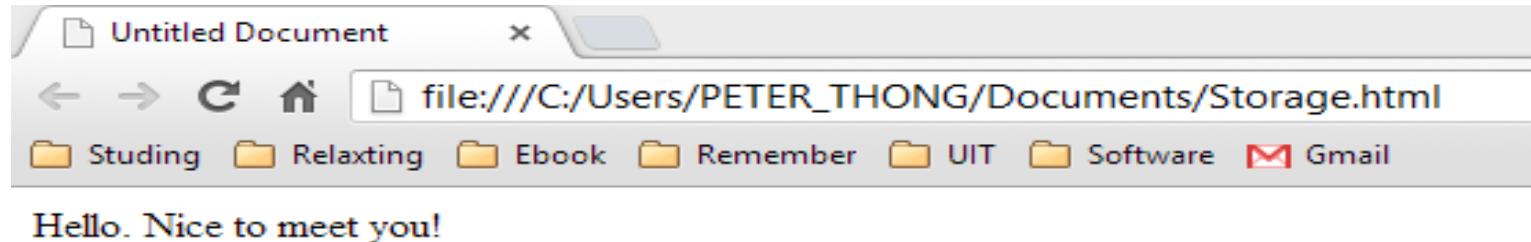


```
<!DOCTYPE HTML>
<html>
<body>
<script type="text/javascript">
    sessionStorage.myVariable="Hello. ";
    localStorage.myVariable="Nice to meet you!";

    document.write(sessionStorage.myVariable);
    document.write(localStorage.myVariable);
</script><body>
</html>
```



# Web Storage(TT)



A screenshot of the Chrome DevTools Elements tab. On the left, there is a tree view of storage types: Frames, Web SQL, indexedDB, Local Storage (which is expanded and has "file://" selected), Session Storage, Cookies, Local Files, and Application Cache. A red oval highlights the "Local Storage" section. To the right is a table with two columns: "Key" and "Value". There is one entry: "myVariable" with the value "Nice to meet you!".

Key	Value
myVariable	Nice to meet you!

# Web Storage(TT)

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. On the left, the Element Tree displays a file named 'Storage.html'. A red arrow points from the 'Session Storage' entry in the tree to the table on the right. The table has two columns: 'Key' and 'Value'. One row shows a key 'myVariable' with a value 'Hello.'.

Key	Value
myVariable	Hello.

Below the table, the page content area displays the text "Hello. Nice to meet you!".

# Nội dung

---



- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Hoạt động nhóm

---



- ◆ Các nhóm xác định sơ đồ ERD cho website bán hàng qua mạng
- ◆ Các nhóm lên bảng thuyết minh sơ đồ ERD của nhóm



# Nội dung

---



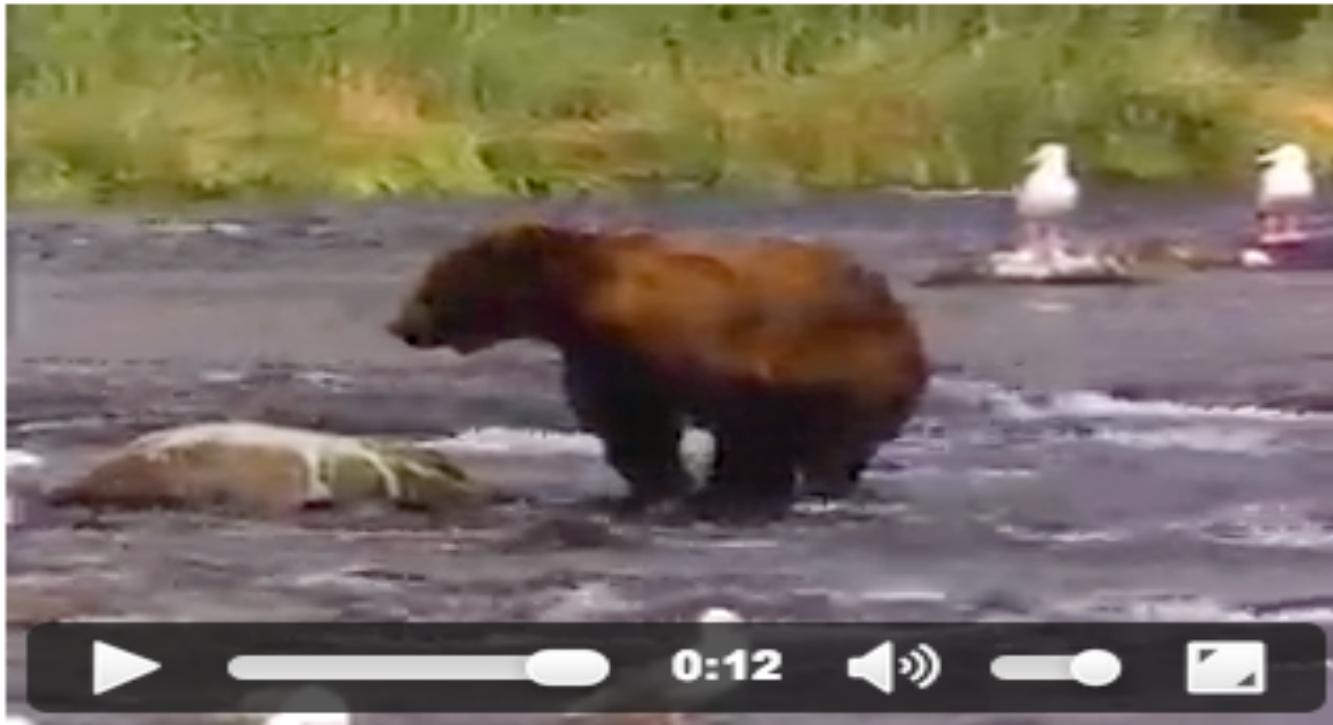
- ★ New Structural Element
- ★ Form Element
- ★ Graphics (Canvas, SVG, CSS3 2D/3D)
- ★ Multimedia (Video Tag, Audio Tag)
- ★ Form Attributes
- ★ New input type
- ★ Web Storage
- ★ Hoạt động nhóm
- ★ Bài tập



# Bài tập



- ◆ **Bài 1:** Dùng tag <video> trong html5 để mở một file .mp4 tùy chọn



# Bài tập (tt)



- ◆ **Bài 2:** Dùng tag <audio> trong html5 để một file .mp3



- ◆ **Bài 3:** Dùng các tag <table>, <form>, <p>, <hr> để thiết kế trang web sau

[Đăng nhập hệ thống](#)

---

Tên đăng nhập	<input type="text" value="nhập tên"/>
Mật khẩu	<input type="password" value="nhập mã số"/>
<input type="button" value="Đăng nhập"/>	
<a href="#">Quên mật khẩu / Đăng ký</a>	

---



# Bài tập (tt)

- ◆ **Bài 4:** Thiết kế trang web như sau:

## Đăng ký tài khoản đăng nhập hệ thống

Họ	Tên
Nhập mật khẩu	
Nhập lại mật khẩu	
Ngày tháng năm sinh mm / dd / yyyy	
Tỉnh thành	
Giới tính	<input type="radio"/> nam
	<input type="radio"/> Nữ
<b>Đăng ký</b>	

# Bài tập (tt)

- ◆ **Bài 5:** Dùng CSS và HTML5 để thiết kế trang Web có các thành phần `<header>`, `<nav>`, `<footer>`, `<aside>` như sau

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Trang chủ Thông tin Giới thiệu Tuyển sinh Nghiên cứu

**Chương trình tiên tiến**  
Chương trình đào tạo tiên tiến" là dự án quốc gia do Bộ GD&ĐT  
Khoa HTTT

**Ngành An Toàn Thông Tin**  
Chương trình đào tạo kỹ sư tài năng ngành An toàn Thông tin cho phép sinh viên tiếp cận với một chương trình đào tạo tiên tiến, thực tập với những công nghệ hiện đại, được sử dụng những tài liệu mới cập nhật và học tập trong điều kiện chất lượng cao.  
Khoa MMT

THÔNG TIN MỚI NHẤT VỀ KỲ THI THPT QUỐC GIA 2015  
Sáng 10/12, Phó Cục trưởng Cục Khảo thí&Kiểm định chất lượng, Bộ GD-ĐT Trần Văn Nghĩa tiếp tục giải đáp hàng loạt câu hỏi nóng liên quan đến kỳ thi THPT quốc gia năm 2015.

Copyright 2014





---

# End!

