



THỰC HỌC – THỰC NGHIỆP

THIẾT KẾ WEB VỚI HTML5 & CSS3

XÂY DỰNG ĐỒ HỌA TRONG HTML5

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

- ⊙ Hiểu được cơ bản về Canvas
- ⊙ Sử dụng canvas để vẽ hình
- ⊙ Hiểu được cơ bản về SVG
- ⊙ Sử dụng SVG để vẽ hình



- 📖 Tổng quan về Canvas
- 📖 Làm việc với Canvas
- 📖 Tổng quan về SVG
- 📖 Làm việc với SVG





PHẦN 1

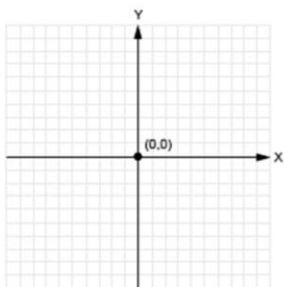
TỔNG QUAN CANVAS

- ☐ Phần tử <canvas> được Apple giới thiệu lần đầu tiên vào năm 2004. Sau đó nó được chuẩn hóa trên nhiều trình duyệt khác nhau.
- ☐ Thẻ <canvas> trong HTML5 được sử dụng để vẽ đồ họa một cách nhanh gọn thông qua một ngôn ngữ nào đó (thông thường là JavaScript).
- ☐ Tuy nhiên, thẻ <canvas> bản thân nó không có khả năng tự vẽ mà nó chỉ là phần bao ngoài cho đồ họa – bạn cần phải sử dụng Javascript để vẽ đồ họa bên trong nó.
- ☐ Canvas sử dụng một số phương thức để vẽ: đường kẻ, hình hộp, hình tròn, chữ và chèn ảnh.

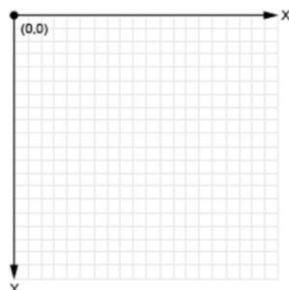
- Như bạn có thể thấy, SVG được sử dụng ở hầu hết mọi nơi và trong vô số tình huống. Tin tốt về tất cả những điều này là hỗ trợ trình duyệt cho SVG đang trở nên tốt hơn.



- Gốc tọa độ được nằm ở góc trên bên trái với trục x mở rộng sang bên phải và Y sẽ mở rộng xuống dưới.



1. Cartesian coordinate space



2. Canvas coordinate space



PHẦN 2

LÀM VIỆC VỚI CANVAS

- ❑ Cần một trình duyệt và phần mềm viết code
- ❑ Có thể đặt bất kì đâu trong <body>
- ❑ Bắt đầu bằng thẻ mở <canvas> và kết thúc là thẻ đóng </canvas>

```
<h3>Giới thiệu về Canvas</h3>
<canvas id="myCanvas" width="500" height="300">
  <p>Trình duyệt của bạn không hỗ trợ HTML5 Canvas.</p>
</canvas> thông báo trên sẽ hiển thị nếu trình duyệt không hỗ trợ Canvas-->
```

❑ Sau đó phần Javascript khai báo như sau:

```
<script>
  var canvas = document.getElementById('myCanvas'),
      context = canvas.getContext('2d');
  // Code ví dụ sẽ tiếp tục ở đây...
</script>
```

- ❖ Dòng 1: Ta tạo biến lưu trữ phần tử canvas thông qua ID
- ❖ Dòng 2: Tạo một biến 'context' trỏ đến 2D context cho canvas thông qua hàm getContext(). Biến này sẽ được dùng để truy cập đến tất cả hàm và thuộc tính của canvas.

Một số thuộc tính của phần tử <canvas>:

- ❑ **fillStyle**: Thuộc tính dùng để thiết lập màu, dải màu hoặc mẫu họa tiết để đổ lên hình
- ❑ **strokeStyle**: Thuộc tính dùng để thiết lập màu sắc, dải màu hoặc mẫu họa tiết sử dụng cho đường viền
- ❑ **lineWidth**: Thuộc tính dùng để thiết lập độ rộng của đường kẻ giá trị được tính bằng pixel

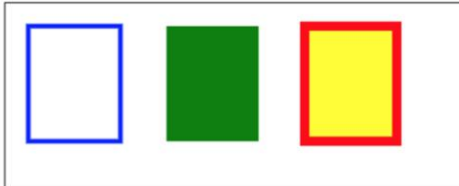
```
<canvas id="myCanvas"
  width="500"
  height="200"
  style="border:1px solid #000;">
</canvas>
<script>
  var theCanvas = document.getElementById("myCanvas");
  var ctx = theCanvas.getContext("2d");
  ctx.fillStyle = 'green';
  ctx.fillRect(20, 20, 100, 100);
  ctx.strokeStyle = "rgba(0,0,255,1)";
  ctx.lineWidth = 5;
  ctx.strokeRect(20,20,100,100);
</script>
```



Một vài phương thức hỗ trợ vẽ hình chữ nhật của <canvas>

- ❑ **strokeRect(x,y,w,h):** Hàm dùng để vẽ một hình chữ nhật (không đổ màu).
- ❑ **fillRect(x,y,w,h):** Hàm fillRect() sử dụng để vẽ một hình chữ nhật đã được tô màu.

```
<canvas id="myCanvas"
  width="500"
  height="200"
  style="border:1px solid #000;">
</canvas>
<script>
  var theCanvas = document.getElementById("myCanvas");
  var ctx = theCanvas.getContext("2d");
  // vẽ hình chữ nhật có đường viền
  ctx.strokeStyle = "blue";
  ctx.lineWidth = 5;
  ctx.strokeRect(25,25,100,125);
  // vẽ hình chữ nhật được đổ màu
  ctx.fillStyle = "green";
  ctx.fillRect(175,25,100,125);
  // Vẽ hình chữ nhật có đổ màu và đường viền
  ctx.strokeStyle = "red";
  ctx.fillStyle = "yellow";
  ctx.lineWidth = 10;
  ctx.fillRect(325,25,100,125);
  ctx.strokeRect(325,25,100,125);
</script>
```



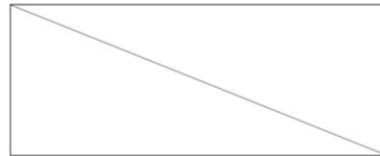
Để vẽ đường thẳng trên một đối tượng canvas, ta sẽ dùng 2 hàm sau:

❑ **moveTo(x,y)**: xác định điểm bắt đầu

❑ **lineTo(x,y)** : xác định điểm kết thúc

Ví dụ: Định điểm bắt đầu (0,0) và điểm kết thúc là (500,200). Sau đó dùng hàm stroke() vẽ một đường thẳng

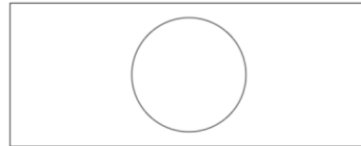
```
<canvas id="myCanvas"
  width="500"
  height="200"
  style="border:1px solid #000;">
</canvas>
<script>
  var theCanvas = document.getElementById("myCanvas");
  var ctx = theCanvas.getContext("2d");
  ctx.moveTo(0,0);
  ctx.lineTo(500,200);
  ctx.stroke();
</script>
```



Để vẽ một hình tròn trên canvas, chúng ta sử dụng phương thức `arc()`:

- ❑ **`arc(x,y,r,start,stop)`**: Để tạo một vòng tròn với `arc()`: Đặt góc bắt đầu thành 0 và góc kết thúc thành $2 * \text{Math.PI}$. Các tham số `x` và `y` xác định tọa độ `x` và `y` của tâm đường tròn. Tham số `r` xác định bán kính của vòng tròn.

```
<canvas id="myCanvas"
  width="500"
  height="200"
  style="border:1px solid #000;">
</canvas>
<script>
  var theCanvas = document.getElementById("myCanvas");
  var ctx = theCanvas.getContext("2d");
  ctx.beginPath();
  ctx.arc(250,100,80,0,2*Math.PI);
  ctx.stroke();
</script>
```



Để viết một đoạn văn bản, ta chú trọng đến thuộc tính và hàm sau:

- ❑ **font** – xác định kiểu chữ cho văn bản
- ❑ **fillText(text,x,y)** – vẽ chữ trên ô canvas

```
<canvas id="myCanvas" width="200" height="100"
  style="border:1px solid #d3d3d3;">
  Your browser does not support the canvas element.
</canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var ctx = canvas.getContext("2d");
  ctx.font = "30px Arial";
  ctx.fillText("Hello World", 10, 50);
</script>
```



Hello World

Một vài thư viện hữu ích hỗ trợ vẽ canvas:

- ❑ [Paper.js](#)
- ❑ [EaselJS](#)
- ❑ [Fabric.js](#)
- ❑ [oCanvas](#)

Với việc sử dụng thư viện canvas, ta đã có thêm một tùy chọn nữa trong việc vẽ đồ thị, hình động hoặc cao hơn là viết các game nhỏ.



PHẦN 2

TỔNG QUAN VỀ SVG

- ❑ SVG là viết tắt của **S**calable **V**ector **G**raphics, nó được phát hành từ những năm 1999, nhưng cho đến năm 2011 mới trở nên phổ biến trên Internet Explorer 9.
- ❑ SVG là một ngôn ngữ dạng XML, dùng để miêu tả hình ảnh đồ họa vector 2 chiều, tĩnh và hoạt hình, dành cho những ứng dụng trên các trang mạng.
- ❑ SVG rất được giới đồ họa ưu chuộng vì nó không phụ thuộc vào độ phân giải, hình có thể phóng to gấp nhiều lần mà không lo "vỡ".
- ❑ Trong HTML5, bạn có thể thêm trực tiếp SVG vào trang HTML.
- ❑ SVG trở thành một W3C Recommendation từ 14/1/2003 và bạn có thể kiểm tra phiên bản mới nhất của nó tại: SVG Specification.

- Đồ họa vector có thể mở rộng

- Tại thời điểm này, hầu hết các trình duyệt web hiện đại đều hỗ trợ các tính năng cơ bản và quan trọng nhất của SVG. Tại caniuse.com, bạn có thể tự kiểm tra hỗ trợ SVG hiện tại.





PHẦN 3

LÀM VIỆC VỚI SVG

- ❑ Hình ảnh SVG có thể được trình duyệt hiển thị bằng cách đưa chúng vào thẻ `img`:

```
<body>
  <!-- Sử dụng SVG -->
  
  <!-- Tương tự như các định dạng ảnh khác -->
  
  
  
  
</body>
```

- ❑ HTML5 cho phép nhúng SVG trực tiếp bằng sử dụng thẻ `<svg>...</svg>` có cú pháp đơn giản tiếp sau:

```
<body>
  <svg width="10" height="10">
    <rect x="0" y="0" width="10" height="10" fill="blue" />
  </svg>
</body>
```

- ❑ SVG có một số yếu tố hình dạng được xác định cụ thể bởi nhà phát triển thông qua các phần tử:
 - ❖ <circle>: xác định để vẽ hình tròn
 - ❖ <rect>: xác định để vẽ hình chữ nhật
 - ❖ <ellipse>: xác định để vẽ hình ellipse
 - ❖ <line>: xác định để vẽ đường kẻ
 - ❖ <polyline>: xác định để vẽ chuỗi đa giác
 - ❖ <poly>:gon>: xác định để vẽ hình Đa giác
 - ❖ <text> xác định để vẽ đoạn văn bản

- ❑ Phần tử <circle> sử dụng để vẽ hình tròn

```
<h2>vẽ đường tròn trong SVG</h2>

<svg id="svgelem" height="200">
  <circle id="redcircle" cx="50" cy="50" r="50" fill="red" />
</svg>
```

vẽ đường tròn trong SVG



- ❑ Giải thích

- ❖ Các thuộc tính **cx** và **cy** xác định tọa độ điểm x và y của tâm của vòng tròn.
- ❖ Nếu **cx** và **cy** bị bỏ qua, tâm của vòng tròn mặc định được đặt thành (0,0)
- ❖ Thuộc tính **r** xác định bán kính của vòng tròn

- ❑ Phần tử `<rect>` sử dụng để tạo hình chữ nhật

```
<svg width="400" height="180">
  <rect x="50" y="20" width="150" height="150"
    style=" fill:blue;
           stroke:pink;
           stroke-width:5;
           fill-opacity:0.1;
           stroke-opacity:0.9" />
</svg>
```



- ❑ Thuộc tính **width** and **height** định chiều rộng và cao hình chữ nhật
- ❑ Thuộc tính **style** sử dụng để định nghĩa CSS cho hình chữ nhật
- ❑ Thuộc tính css **fill** dùng để phủ màu lên hình chữ nhật
- ❑ Thuộc tính css **stroke-width** dùng để định nghĩa chiều rộng của viền
- ❑ Thuộc tính css **stroke** dùng để định nghĩa màu cho đường viền

- ❑ Phần tử <line> dùng để tạo một đường kẻ:

```
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200"
        style="stroke:rgb(255,0,0);stroke-width:2" />
</svg>
```

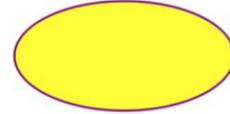


❑ Giải thích

- ❖ Thuộc tính **x1** xác định điểm bắt đầu của đường kẻ theo trục x
- ❖ Thuộc tính **y1** xác định điểm bắt đầu của đường kẻ theo trục y
- ❖ Thuộc tính **x2** xác định điểm kết thúc của đường kẻ theo trục x
- ❖ Thuộc tính **y2** xác định điểm kết thúc của đường kẻ theo trục y.

- ❑ Phần tử <ellipse> được sử dụng để vẽ hình ellipse
- ❑ Code ellipse gần giống với vẽ hình tròn, nhưng khác biệt ở chỗ ellipse có bán kính x và y khác nhau, trong khi hình tròn có bán kính x và y bằng nhau

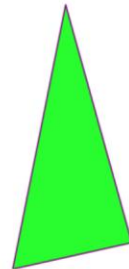
```
<svg height="140" width="500">  
  <ellipse cx="120" cy="80" rx="100" ry="50"  
    style="fill:yellow;stroke:purple;stroke-width:2" />  
</svg>
```



- ❑ Giải thích
 - ❖ Thuộc tính **cx** xác định tọa độ x của tâm hình ellipse
 - ❖ Thuộc tính **cy** xác định tọa độ y của tâm hình ellipse
 - ❖ Thuộc tính **rx** xác định bán kính ngang
 - ❖ Thuộc tính **ry** xác định bán kính dọc

- ❑ Phần tử <polygon> sử dụng để tạo đồ họa chứa ít nhất 3 góc
- ❑ Đa giác được tạo thành từ các đường thẳng và hình vẽ phải được nối liền nhau.

```
<svg height="210" width="500">  
  <polygon points="200,10 250,190 160,210"  
    style="fill:lime;stroke:purple;stroke-width:1" />  
</svg>
```



- ❑ Giải thích
 - ❖ Thuộc tính points xác định tọa độ x và y cho mỗi góc của đa giác

- ❑ Phần tử <polyline> được sử dụng để tạo bất kỳ hình dạng nào chỉ bao gồm các đường thẳng (được kết nối tại một số điểm):

```
<svg height="200" width="500">
  <polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
    style="fill:none;stroke:black;stroke-width:3" />
</svg>
```



- ❑ Giải thích

- ❖ Thuộc tính points xác định danh sách các điểm (cặp tọa độ x và y) cần thiết để vẽ chuỗi đa giác

SỰ KHÁC NHAU GIỮA SVG VÀ CANVAS

HTML5 giới thiệu hai thành phần đồ họa là Canvas và SVG nhưng chúng lại có nhiều điểm khác nhau. Hiểu được sự khác nhau đó sẽ giúp các bạn sử dụng chúng hiệu quả hơn.

CANVAS	SVG
Phụ thuộc vào độ phân giải màn hình	Không phụ thuộc vào độ phân giải màn hình
Không hỗ trợ xử lý sự kiện	Hỗ trợ xử lý sự kiện
Không trở thành 1 phần của DOM	Trở thành 1 phần của DOM
Chỉ chỉnh sửa qua script	Có thể chỉnh sửa qua script và css
Kéo giãn thì giảm chất lượng hình	Kéo giữ vẫn giữ chất lượng tốt
Dựa trên pixel	Dựa trên vector

- ☑ Tổng quan về Canvas
- ☑ Làm việc với Canvas
- ☑ Tổng quan về SVG
- ☑ Làm việc với SVG



thank
you!