Introduction to web technologies and HTML 5

HTML 5 Graphics

ITI – Assiut Branch

Eng. Hany Saad

Graphics

SVG

- SVG stands for Scalable Vector Graphics and it is a language for describing 2D-graphics and graphical applications in XML
- SVG is W3C standard
- HTML5 allows embedding SVG directly using

```
<svg>...</svg>
```

Graphics (Cont.)

- **SVG would draw**
  - rectangle using

```
<rect x="" y="" width="“" height="“" style="“”>
```

  - line using

```
<line x1="“" y1="“” x2="“" y2="““ style="“”>
```

  - circle using

```
<circle cx="" cy="“" r="“" stroke="““ stroke-
width="“" fill="““">
```

  - ellipse using

```
<ellipse cx="“" cy="“" rx="“" ry=""“ style="“”>
```

Graphics (Cont.)

- path

  <path d=“”>

- polygon using

  <polygon points=“”> tag

- polyline using

  <polyline points=“”> tag

Graphics (Cont.)

- **Canvas**
  - Canvas is a new HTML element which can be used to draw graphics on a web page using Javascript.
  - A canvas is a rectangular area, that you control every pixel of it.
  - The canvas element has several methods for drawing paths, boxes, circles, characters, and adding images.
  - <canvas> element is an HTML tag, with the exception that its contents are rendered with JavaScript.
  - It creates a fixed size drawing surface that exposes one or more rendering contexts using canvas context object.
  - Each canvas element can only have one context that can be “2d”.

Graphics (Cont.)

- Draw dynamic and interactive graphics.
- Draw images using 2D drawing API.
  - Lines, curves, paths, shapes, fill styles, etc.
- Useful for:
  - Graphs.
  - Applications.
  - Games and Puzzles.
  - And more…

Graphics (Cont.)

- **Steps to follow**
  - Place the canvas tag somewhere inside the HTML document,
  - Access the canvas tag with JavaScript,
  - Create a 2D context, and then
  - Utilize the HTML5 Canvas API to draw visualizations.

```
<canvas id="myCanvas" width="578“ height="200">
</canvas>
<script>
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');
// do stuff here
 </script>
```

Graphics (Cont.)

- **Canvas Element & Canvas Context**
  - The canvas element is the actual DOM node that's embedded in the HTML page.
  - The canvas context is an object with properties and methods that you can use to render graphics inside the canvas element.
  - The context is 2d.

Graphics (Cont.)

- **Canvas Context Properties & Methods**
  - Line
  - Curve
  - Path
  - Shapes
    - Rectangle
    - Circle
    - Custom Shapes
  - Fill Styles
  - Text
  - Images

Graphics (Cont.)

- **Line using HTML5 CanvasLine**
  - To draw a line using HTML5 Canvas
    - First, use the beginPath()
      - method to declare that we are about to draw a new path.
    - Next, use the moveTo()
      - method to position the context point (i.e. drawing cursor
    - Then, use the lineTo()
      - method to draw a straight line from the starting position to a new position.
    - Finally, to make the line visible, we can apply a stroke to the line using stroke().
    - Note: without declaring strokeStyle property before using stroke(), the stroke default color is black

Graphics (Cont.)

- **Line useful Properties & Methods**
  - lineWidth
    - used to define width of the required line to be drawn in px,
    - should be declared before strokeStyle property.
  - lineCap = square | round | butt
    - declares how the drawn line ends look
  - lineJoin = bevel | round | miter
    - declares how two lines are joined together

Graphics (Cont.)

❑ Curves & Arcs Using HTML5 Canvas

```
arc(x, y, radius, startAngle, endAngle, antiClockwise );
```

- An arc is nothing more than a section of the circumference of an imaginary circle that can be defined by x, y, and radius.
- startAngle and endAngle. These two angles are defined in radians.
- antiClockwise which defines the direction of the arc path between its two ending points, its default is false
  - i.e. the arc to be drawn is clockwise

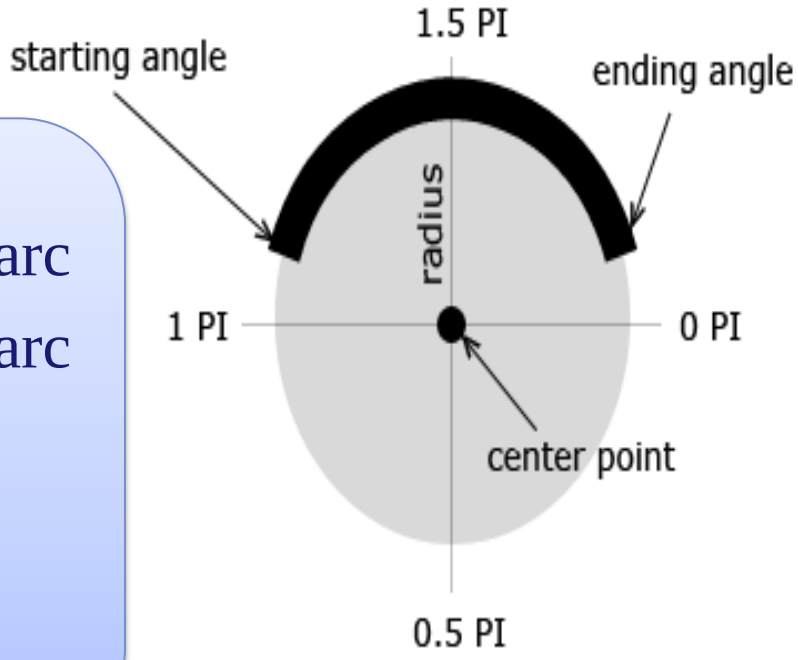
Graphics (Cont.)

❑ Curves & Arcs Using HTML5 Canvas

arc(x, y, radius, startAngle, endAngle, antiClockwise);

- Example:

```
arc(
210, // X coordinate of the start of arc
210, // Y coordinate of the start of arc
200, // Radius
0, // Start angle
Math.PI * 2, // End angle
true); // Anticlockwise
```



Graphics (Cont.)

- **Circle & Semi-Circle using HTML5 Canvas**
  - To draw a circle
    - Use arc() method and define its starting angle as 0 and the ending angle as 2 * PI.

```
arc(x, y, radius, 0, 2*Math.PI, anticlk);
```

  - To draw a semi-circle
    - Use arc() method and define its ending angle has startAngle + PI.

```
arc(x, y, radius, sAngle, sAngel+Math.PI, anticlk);
```

Graphics (Cont.)

- **Rectangle using HTML5 Canvas**

```
rect(x, y, width, height)
fillRect(x, y, width, height)
strokeRect(x, y, width, height)
clearRect(x, y, width, height)
```

  - An HTML5 Canvas rectangle is positioned with x and y parameters, and is sized with width and height parameters.
  - The rectangle is positioned about its top left corner.

Graphics (Cont.)

- **Paths & shapes using HTML5 Canvas**
  - To create a path with HTML5 Canvas, connect multiple subpaths using
    - lineTo(),
    - arcTo(),
  - To create a custom shape
    - First create a path and mentioned above
    - Then, close it using the closePath()
  - Note:
    - beginPath() is used in the beginning to start drawing a new path.
    - fillStyle property & fill() can be used to fill in color within drawn shape.

Graphics (Cont.)

- **Gradient**
  - Gradient can be used to fill rectangles, circles, lines, text, etc..
  - Linear Gradient

```
createLinearGradient(startX, startY, endX, endY);
```

  - Radial Gradient

```
createRadialGradient(startX, startY, startRadius, endX, endY, endRadius);
```

  - Note:

```
addColorStop(offset, color);
```

    - It can be called multiple times to change a gradient
    - Its offset value between 0.0 and 1.0

Graphics (Cont.)

- **Text Properties & Methods**
  - Font
    - style, size, font family
  - fillStyle
    - color or rgb()
  - fillText(txt, x, y)
  - strokeStyle
    - color or rgb()
  - strokeText(txt, x, y)

Graphics (Cont.)

❑ Dealing with ImageFont

- To draw an image on canvas area we use
  - drawImage(imgObj, x, y [, width, height])
    - ➢ imgObj defines image required to be displayed, it must be created first and wait for being loaded befor instantiating drawImage().
    - ➢ x,y defines top left corner of the image relative to the top left corner of the canvas
    - ➢ width, height define width, height of the displayed image
  - Note:
    - ➢ Construct your image object using “new Image()”

Graphics (Cont.)

- **Scaling, Rotating & Translating**
  - scale(x, y)
    - resize current drawing either bigger or smaller
  - rotate(angle)
    - rotate the current context around the origin within the canvas area
  - translate(x, y)
    - move current context within the canvas area into a different point

Self Study

❑ Self Study Topics:

- **Other HTML5 APIs:**
  - HTML5 Drag & Drop
  - HTML App cache
  - HTML Web Workers
  - HTML SSE (Server Sent Events).

- **Other Curves Types:**
  - quadraticCurve
  - bezierCurve

Summery..

<Questions> ? </Questions>

Thank You…