

Client-Side Graphics with: Drawing and styling lines and shapes.

Client-Side Graphics with The canvas element is used to render simple graphics such as line art, graphs, and other custom graphical elements on the client side.

From a markup point of view, there is little that you can do with a tag. You simply put the element in the page, name it with an id attribute, and define its dimensions with height and width attributes:

```
<canvas id="canvas" width="300" height="300">
  <strong>Canvas Supporting Browser Required</strong>
</canvas>
```

Canvas Supporting Browser Required Note the alternative content placed within the element for browsers that don't support the element. After you place a tag in a document, your next step is to use JavaScript to access and draw on the element. For example, the following fetches the object by its id value and creates a two-dimensional drawing context:

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

Draw lines

```
context.beginPath();
context.lineTo(20,100);
context.lineTo(120,300);
context.lineTo(220,100);
context.stroke();
```



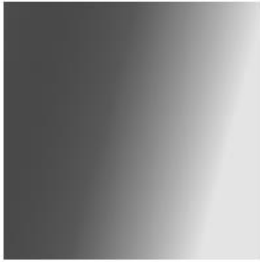
Draw Triangle

```
context.strokeStyle = "blue";
context.fillStyle = "red";

context.lineWidth = 10;
context.beginPath();
context.lineTo(200,10);
context.lineTo(200,50);
context.lineTo(380,10);
context.closePath();
context.stroke();
context.fill();
```



```
var lg = context.createLinearGradient(10,150,200,200);  
lg.addColorStop(0,"#B03060");  
lg.addColorStop(0.5,"#4169E1");  
lg.addColorStop(1,"#FFE4E1");  
context.fillStyle = lg;  
context.beginPath();  
context.rect(10,150,200,200);  
context.fill();
```



```
var rg = context.createRadialGradient(350,300,80,360,250,80);  
rg.addColorStop(0,"#A7D30C");  
rg.addColorStop(0.9,"#019F62");  
rg.addColorStop(1,"rgba(1,159,98,0)");  
context.fillStyle = rg;  
context.beginPath();  
context.fillRect(250,150,200,200);
```



1. **Write Javascript code to draw rectangle and square shape using canvas.**

```
<!DOCTYPE html>
<html>
<head>
<title>HTML5 canvas example</title>
<script type="text/javascript">

window.onload = function() {
  var canvas = document.getElementById("canvas");
  var context = canvas.getContext("2d");
  context.strokeStyle = "orange";
  context.strokeRect(10,10,150,50);
  context.fillStyle = "rgba(218,0,0,0.4)";
  context.fillRect(150,30,75,75);
}
</script>
</head>
<body>
<h1>Simple Canvas Examples</h1>
<canvas id="canvas" width="300" height="300">
  <strong>Canvas Supporting Browser Required</strong>
</canvas>
</body>
</html>
```

Output:

Simple Canvas Examples



2. Write JavaScript code to draw various shapes like line, triangle, square and rectangle using canvas.

```
<!DOCTYPE html>
<html> <head>
<title>HTML5 canvas lines and shapes example</title>
<script type="text/javascript">
window.onload = function() {
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext("2d");
    context.strokeStyle = "blue";
    context.fillStyle = "red";
    context.lineWidth = 10;
    context.beginPath();
    context.lineTo(200,10);
    context.lineTo(200,50);
    context.lineTo(380,10);
    context.closePath();context.stroke(); context.fill();
    var lg = context.createLinearGradient(10, 150, 200, 200);
    lg.addColorStop(0, "#B03060");lg.addColorStop(0.5, "#4169E1");lg.addColorStop(1, "#FFE4E1");
    context.fillStyle = lg;
    context.beginPath();context.rect (10, 150, 200, 200);context.fill();
    var rg = context.createRadialGradient(50,50,10,60,60,50);
    rg.addColorStop(0, "#A7D30C");
    rg.addColorStop(0.9, "#019F62");
    rg.addColorStop(1, "rgba(1,159,98,0)");
    context.fillStyle = rg;
    context.beginPath();
    context.fillRect(0,0,130,230);
    context.beginPath();
    context.lineTo(250,150);context.lineTo(330,240); context.lineTo(410,150); context.stroke();
}
</script></head> <body>
<h1>Simple Shapes on canvas Example</h1>
<canvas id="canvas" width="500" height="500">
    <strong>Canvas Supporting Browser Required</strong>
</canvas> </body> </html>
```

Output:

Simple Shapes on canvas Example

