

Programming - Lektion 7

...

PC 231006

Mer om att rita på en Canvas

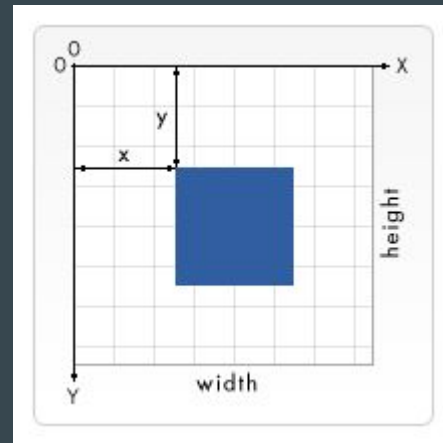
För att rita på en canvas i JavaScript används något som kallas “path” eller banor på svenska

Att rita en path kan liknas med att vi tar upp en penna i en viss färg och tjocklek och ritar med den pennan.

Vill vi sedan ta en ny färg behöver vi starta en ny path.

Koordinatsystemet

Som vi redan sett så är y-axeln riktad nedåt i koordinatsystemet på en canvas.



Funktioner för en path

`beginPath()` - skapar en ny path.

`closePath()` - avslutar en path

`stroke()` - ritar linjerna i en path, inget syns på skärmen förrän stroke (eller fill) anropas.

`fill()` - fyller en path med en färg.

`moveTo(x, y)` - flyttar startpunkten för en path eller del av en path till (x,y)

Funktioner för att rita

`lineTo(x, y)` - ritar en linje från den aktuella punkten till punkten (x, y).

`arc(x, y, radius, startAngle, endAngle)` - Ritar en cirkelbåge, vinklarna anges i radianer.

`rect(x, y, width, height)` - ritar en rektangel.

`fillRect(x, y, width, height)` - Ritar en fylld rektangel.

Det finns funktioner för att rita ellipser, rektanglar med rundade hörn med mera. Vi kan även rita mjuka banor, så kallade Bezierkurvor, se [dokumentationen](#)

Bestämma färg, tjocklek på linje och fyllningsfärg

Egenskaperna hos en path bestäms av ett antal variabler:

```
strokeStyle = "red"
```

```
lineWidth = 15
```

```
fillStyle = "blue"
```

Text på Canvas

Det är inte lika enkelt att skriva text som med hjälp av simple.js.

Text ses som en fylld figur, så först måste jag välja fyllningsfärg, sedan väljer jag typsnitt och slutligen skriver jag ut min text.

```
//Skriv en text  
context.fillStyle= "Gold"  
context.font = "34px Arial"  
context.fillText( "Hej!", 100, 110)
```

En minimal html-fil

Vår canvas och JavaScript måste ligga på en html-sida. En minimal sida kan se ut så här.

```
<body>
  <canvas id="minCanvas" width="400" height="400"></canvas>

  <script>
    // Här ligger mitt program
  </script>
</body>
```

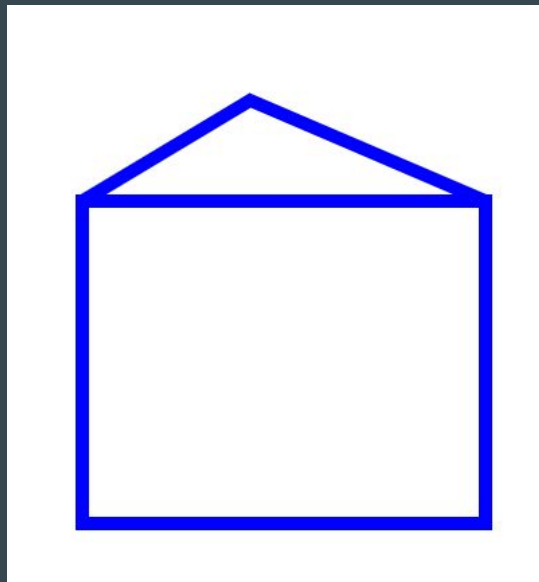

Vi behöver komma åt canvas-elementet från vår kod

Canvas elementet är en del av vår websida. För att nå canvasen från vår kod behöver vi alltid de följande raderna i vårt script

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

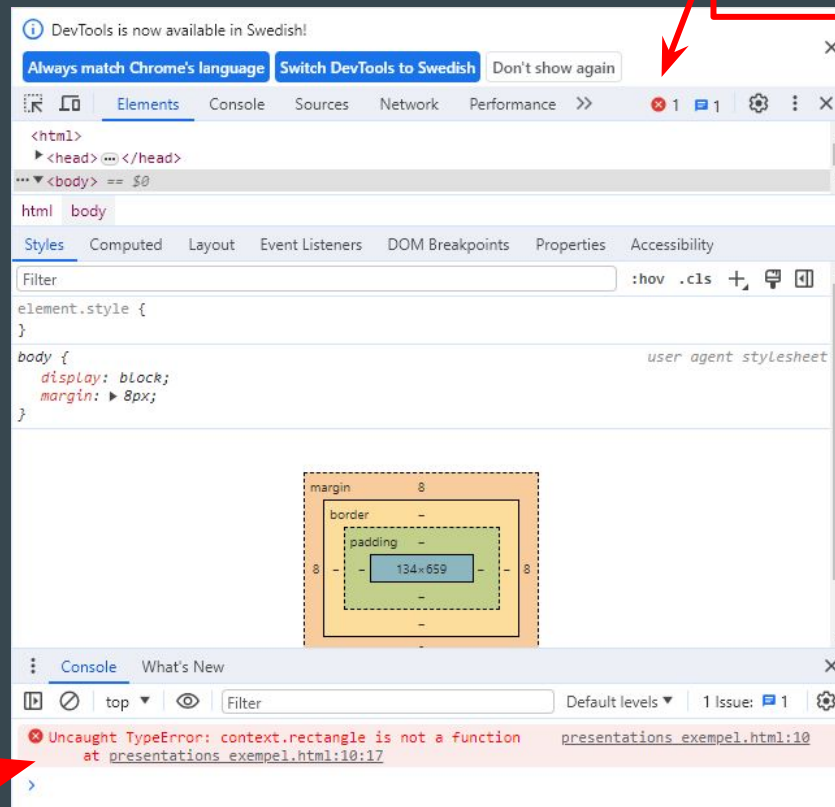
Ett exempel

```
<script>
    var canvas = document.getElementById("minCanvas");
    var context = canvas.getContext("2d");
    context.beginPath();
    context.strokeStyle="blue";
    context.lineWidth = 10;
    context.rect(75, 150, 300, 240);
    context.moveTo(75, 150);
    context.lineTo(200,75);
    context.lineTo(375, 150);
    context.lineTo
    context.stroke();
    context.closePath();
</script>
```



Felsökningstips

I Chrome kan du trycka F12 eller högerklicka och välj "Inspect" för att se utvecklarverktyget. Det innehåller mycket som vi inte förstår (och som vi inte kommer lära oss i kursen) men kan vara en god hjälp när vårt script inte fungerar som väntat. I det här exemplet har jag skrivit fel på kommandot för att rita en rektangel.



Felaktig kod

Här är en kod med några misstag, låt oss felsöka den i utvecklarläget.

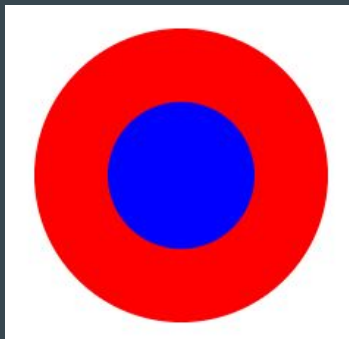
```
let name = "Peter"  
for (let i =1 ; i<5, i++){  
    name= namr  +" " +i;  
}  
alert(name);
```

Tryck

- F12
- Ctrl/Command + Shift + I
- Högerklicka och välj
“Inspect”

Att rita med olika färger.

Om jag ritat något i en färg och sedan vill rita något i en annan färg så måste jag skapa en ny path. Se exemplet.

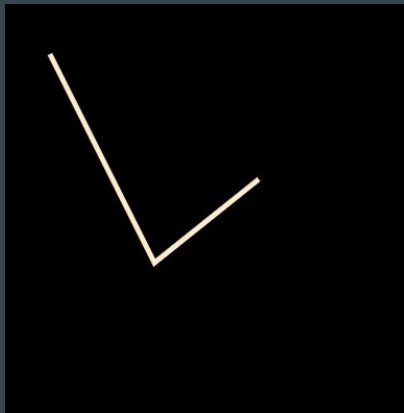


```
context.fillStyle = "red"
context.beginPath();
context.arc(200,200,100, 0, 2*Math.PI);
context.fill();
context.closePath();
context.beginPath();
context.fillStyle="blue"
context.arc(200,200,50, 0, 2*Math.PI);
context.fill();
context.closePath();
```

Att fylla en figur med färg.

Om jag vill skapa en geometrisk figur som ska fyllas med färg behöver jag inte rita alla kanter.

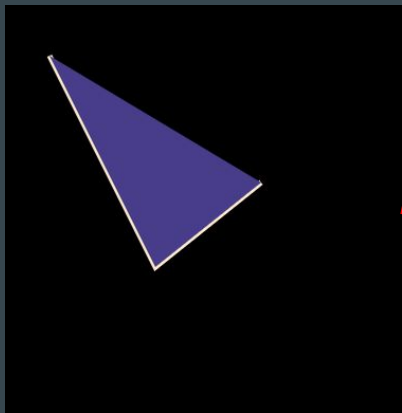
Här ritar jag två linjer



```
context.beginPath();  
context.strokeStyle = "BlanchedAlmond";  
context.fillStyle = "DarkSlateBlue";  
context.lineWidth = 5;  
context.moveTo(50, 50);  
context.lineTo(150,250);  
context.lineTo(250, 170);  
// Vi måste anropa stroke för att rita något på skärmen  
context.stroke();
```

Att fylla en figur med färg.

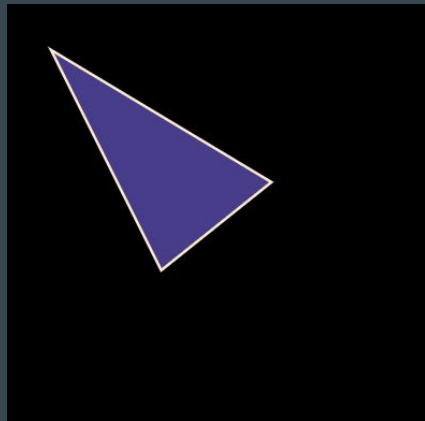
Om jag nu vill fylla min form med en färg så är inte sista sidan nödvändig.



```
context.beginPath();
context.strokeStyle = "BlanchedAlmond";
context.fillStyle = "DarkSlateBlue";
context.lineWidth = 5;
context.moveTo(50, 50);
context.lineTo(150,250);
context.lineTo(250, 170);
// Vi måste anropa stroke för att rita något på skärmen
context.stroke();
context.fill();
```

Att stänga figuren

Om jag vill ha den tredje sidan ritad kan jag antingen själv räkna ut linjen som behövs eller använda `closePath()`

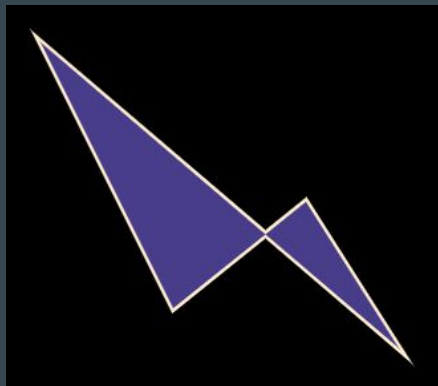


```
context.beginPath();
context.strokeStyle = "BlanchedAlmond";
context.fillStyle = "DarkSlateBlue";
context.lineWidth = 5;
context.moveTo(50, 50);
context.lineTo(150, 250);
context.lineTo(250, 170);
// Vi måste anropa stroke för att rita något på skärmen
context.closePath();
context.stroke();
context.fill();
```


Korsande linjer

Vi kan fylla ytor som har korsande linjer.

JavaScript tolkar att vi vill ta den kortaste vägen tillbaka till början av figuren



```
context.beginPath();
context.strokeStyle = "BlanchedAlmond";
context.fillStyle = "DarkSlateBlue";
context.lineWidth = 5;
context.moveTo(50, 50);
context.lineTo(150,250);
context.lineTo(250, 170);
context.lineTo(320,280);
// Vi måste anropa stroke för att rita något på skärmen
context.closePath();
context.stroke();
context.fill();
```

Lite mer om färger...

Vi har tidigare pratat om att färger anges som RGB, dvs Red-Green-Blue med värden mellan 0 och 255 (0 och FF hexadecimalt). Det går även att göra färger transparenta med hjälp av en så kallat alpha-värde.

För att skapa färger med transparens så kan vi använd `rgba(r,g,b,a)`

Pröva med lite olika värde på `a` mellan 0 och 1.

```
context.beginPath();  
context.fillStyle = "rgba(255,0,0,0.5)";  
context.beginPath();  
context.arc(200,200,100, 0, 2*Math.PI);  
context.fill();  
context.closePath();  
context.beginPath();  
context.fillStyle="rgba(0,0,255,0.5)";  
context.arc(200,200,50, 0, 2*Math.PI);  
context.fill();  
context.closePath();
```